Novel Skeletal Representation for Articulated Creatures

A Thesis
Presented to
The Academic Faculty

by

Gabriel J. Brostow

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

College of Computing
Georgia Institute of Technology
April 2004

Novel Skeletal Representation for Articulated Creatures

Approved by:

Irfan A. Essa, Committee Chair	Leonard McMillan (UNC at Chapel Hill)
Aaron Bobick	James Rehg
Jessica K. Hodgins (Carnegie Mellon University)	Greg Turk

Date Approved: 12 April 2004

TABLE OF CONTENTS

LIST OF TABLES . . . . . **v**

LIST OF FIGURES . . . . . **vi**

SUMMARY . . . . . **x**

CHAPTER I INTRODUCTION **1**

1.1 Broader Impact and Applications 3

1.2 Mathematical Objective 6

CHAPTER II RELATED WORK **9**

CHAPTER III SPINE FORMULATION & ESTIMATION **13**

3.1 Formulation 13

3.2 Creating a Spine for a Single Frame 15

3.3 Correspondence Tracking 20

3.4 Imposing a Single Graph on the Spine 22

CHAPTER IV IMPLEMENTATION & EXPERIMENTAL RESULTS **23**

4.1 Design Decisions 23

4.2 Data Acquisition and Processing 24

4.3 Results 29

4.4 Limitations 41

CHAPTER V COMPARISON OF APPLICABILITY **43**

5.1 Possible Applications Overview 43

5.2 The Motion Capture Problem 45

5.3 Current Solution 45

5.4 Incorporating Spines 48

5.5 Evaluation Plan 48

CHAPTER VI CONCLUSION & FUTURE WORK **51**

6.1 Contributions 51

6.2 Future Work 53
6.2.1 Fusing Spine Nodes 54
6.2.2 Limb Optimization Frameworks 55
6.2.3 Use of this Representation 58
6.3 Final Discussion 59
REFERENCES **60**

LIST OF TABLES

Table 1 Context of our approach (marked in color) with respect to existing algorithms in established fields. 9

LIST OF FIGURES

Figure 1 Photos of five subjects filmed to obtain experimental data. The middle image shows the surface of the adult human, and pictured clockwise from the upper left is the human baby (11 months old), the Rosehair tarantula, the dog, and the camel marionette. 4

Figure 2 Spine graph limbs encoding motion over time; nodes labeled for illustration only. 7

Figure 3 (A) Articulated subject, (B) reconstructed surface, (C) extracted skeleton. 8

Figure 4 Example of generating a skeleton for a synthetic starfish mesh. (A) Capture images of the starfish from a variety of vantage points (B) Extract a 3D surface using generalized voxel carving and improved marching cubes (C) Starting at one extremity tip, calculate geodesic distances for each vertex (D) Quantize distances and cluster vertices into bins of the same distance (E) Create a skeleton by walking through the progression of level set rings (F) Repeat C-E for each tip and merge into a single representative skeleton. 16

Figure 5 2D example of clustering connected vertices into bins of similar geodesic distance and walking through the resulting level set rings. 17

Figure 6 The red and green skeletons represent the same "creature," possibly seeded from two different places. Wishing to copy nodes from the best limbs each constituent skeleton has to offer, we developed a leaf-node seeking topology matching algorithm that recognizes that these pairs of three-way junctions should be a single four-way junction. 18

Figure 7 Refinement through imposing of correspondence into the sequence. Instead of greedily including every protrusion that appears to be an end effector, we are able to keep only the limbs that appear consistently over time. 21

Figure 8 The sequence of skeleton-trees (left) has separate node-branches $L_A..L_I$. The limb-to-limb correspondence is known across time, but each node exists only in one limb for one frame. Normalizing each limb's length with respect to time, we resample the Spine to form one set of Spine-nodes (right) whose position varies as a function of time. 22

Figure 9 Construction previsualization of our data capture stage. The structure is designed to accommodate small subjects, and support 20 or more cameras without obstructing incoming light. Note "floating" calibration pattern of dimensions 20 x 14cm. 25

Figure 10 Data capture stage surrounding a platform with an interchangeable table top. 26

Figure 11 Camel marionette used for experimentation, after replacing strings with fishing line (inset). This segmented example frame from the video footage shows the blue 2D bounding box, and the subsequently estimated bounding volume used by the GVC algorithm. 29

Figure 12 BABY DATASET: From left to right, one of the views, voxels, polygonal model, level sets, and Spine with distance function. 30

Figure 13 DOG DATASET: From left to right, subject, polygonal model, distance function, level sets, and resulting Spine. 30

Figure 14 CAMEL PUPPET DATASET: From left to right, one view, wireframe, distance function, level sets, and resulting Spine. 30

Figure 15 Graph of camel's number of end effector limb tips found over time when processing surfaces individually. $|E_t|$ varies between four and ten. 32

Figure 16 Histogram showing how many surfaces in the camel sequence were found to have each number of limb tips. The camel sequence revealed five limb tips with the greatest consistency, as is appropriate for this creature. . . . 33

Figure 17 Rendered version of the camel Spine, extracted from the animated sequence. Pictured nodes appear consistently in each frame with known correspondence and orientation. Each limb was parameterized on length and discretized into 15 equal samples, though any subdivision can be used. Frames where limbs were tucked are interpolated for animation purposes. 33

Figure 18 Photograph of tarantula subject taken through transparent table top. Beyond eight legs, tarantulas have a pedipalp on each side of the fangs, and an abdomen section. 34

Figure 19 Tiled subimages of tarantula subject's footage filmed with 20 cameras. Variations in colorsare result of user error in adjusting settings, though two images pictured in bottom row come from JVC cameras instead of Canon. 35

Figure 20 Reconstructed surface mesh of tarantula using array of frames #29. {Red, green, blue} coloring represents $\{ x,y,z \}$ components of surface normals. Lumps on surface are result of fine subdivision of voxel volume despite comparatively large pixels in video footage (at maximal zoom). . 36

Figure 21 Geodesic distance measured from first automatically detected tarantula limb tip, colored from black to white with increasing distance. Subdivision of this distance field into connected components of level sets according to our algorithm produces the pictured skeleton. Other limb tips subsequently yield other skeletons that are merged with the pictured one. 37

Figure 22 Different views of the merged tarantula skeleton: (A)Without edges connecting skeleton nodes, (B) with edges that converge on a point. 37

Figure 23 Graph of tarantula's number of end effector limb tips found over time when processing surfaces individually. $|E_t|$ varies between seven and fifteen. 38

Figure 24 Histogram showing how many surfaces in the tarantula sequence were found to have each number of limb tips. The tarantula sequence revealed ten limb tips with the greatest consistency, which is almost appropriate; tarantulas have eight legs, two pedipalps, and an abdomen. 38

Figure 25 Rendering of Spine estimated for adult human subject (left), and corresponding surface mesh colored according to normals. 39

Figure 26 (A) Graph of adult human's number of end effector limb tips found over time when processing surfaces individually. $|E_t|$ varies between two and five. (B) While most data was processed by quantizing the geodesic distance field into 30 levels, the pictured graph shows the change in $|E_t|$ when using 40 levels instead. The number of levels used cannot be arbitrarily high unless triangles in the surface mesh are subdivided, because level sets must form complete circuits around the body. 40

Figure 27 Histogram showing how many surfaces in the adult human sequence were found to have each number of limb tips. The adult human sequence revealed four limb tips with the greatest consistency instead of five, because one of the two arms was alternatingly tucked against the body. . . . 41

Figure 28 Example illustrating difficulty in aligning coordinate frames defined by three or more markers each. Co-locating one red and blue corner of the two frames can easily leave the other two corners grossly misaligned, so best fit techniques must be used instead (*e.g.*, least squared error on distance). 46

Figure 29 Illustration of attaching bone screw markers to a skeleton. Depending on prior knowledge about limb lengths and joint locations with respect to markers, not all limbs need three screws to be tracked effectively. . . . 47

Figure 30 (A) Illustration of explicit optical motion capture markers attached to subject's surface (top). In postprocessing, a user selects clouds of markers that subsequently drive the pose of each limb in a revolute joint armature (bottom). (x,y,z) location of each optical marker is tracked and available for pose estimation (B) Illustration of implicit Spine node markers in subject's interior. In the same manner as for motion capture, a user can select clusters of nodes to drive each of the two pictured limbs on an armature. Spine node markers have (x,y,z) location and absolute orientation. 50

Figure 31 One hypothesized separation of this branch of the Spine tree into three sections, each being evaluated as rigidly rotating about its parent joint. . 56

Figure 32 While both are Spine trees appearing to have three limbs, the creature on the left has three legs, while the one on the right is a snake with bull-horns. Performing a local Spine flexibility analysis would reveal a good starting point for the placement of three joints in either case, and reduces superfluous joint-location hypotheses. 57

Figure 33 Volume Capture merges technologies of data acquisition for skeletal poses (mocap), surface models (Cyberware), and lighting/texture ([26]). . 58

SUMMARY

This research examines an approach for capturing 3D surface and structural data of moving articulated creatures. Given the task of non-invasively and automatically capturing such data, a methodology and the associated experiments are presented, that apply to multi-view videos of the subject's motion. Our thesis states: A functional structure and the time-varying surface of an articulated creature subject are contained in a sequence of its 3D data. A functional structure is one example of the possible arrangements of internal mechanisms (kinematic joints, springs, etc.) that is capable of performing the motions observed in the input data.

Volumetric structures are frequently used as shape descriptors for 3D data. The capture of such data is being facilitated by developments in multi-view video and range scanning, extending to subjects that are alive and moving. In this research, we examine vision-based modeling and the related representation of moving articulated creatures using Spines. We define a Spine as a branching axial structure representing the shape and topology of a 3D object's limbs, and capturing the limbs' correspondence and motion over time.

The Spine concept builds on skeletal representations often used to describe the internal structure of an articulated object and the significant protrusions. Our representation of a Spine provides for enhancements over a 3D skeleton. These enhancements form temporally consistent limb hierarchies that contain correspondence information about real motion data. We present a practical implementation that approximates a Spine's joint probability function to reconstruct Spines for synthetic and real subjects that move. In general, our approach combines the objectives of generalized cylinders, 3D scanning, and markerless motion capture to generate baseline models from real puppets, animals, and human subjects.

CHAPTER I

INTRODUCTION

This research aims to study a new visual representation of real moving subjects that models both their movement and their volumetric measurements from multiple video sources. The primary goal for this research is to develop a spatio-temporal representation of moving volumes and their kinematic structure, from direct observation. This dissertation explores an automatic data-driven modeling technique for articulated creatures in motion. Given the task of non-invasively and automatically capturing surface and structural data, we present an algorithmic approach and the associated experiments that apply to multi-view videos of the moving subject. To model a moving articulated creature, we treat its body as an assembly of component limbs. The creature's motion is then defined in terms of those limbs, *i.e.,* the changing pose. To attain these goals, we explore the following:

Structure: Extracting a hierarchical limb structure from volumetric data;

Correspondence: Tracking the location of each limb throughout a sequence;

Parameterization: Synthesizing a representation of each limb's shape over time.

The decision to pursue our goals using only data-driven structure and correspondence bears explanation. In contrast to system building where one can and should leverage prior information and situation specific heuristics, we seek a general approach. Our approach relies on bottom-up data-driven analysis, aiming only to derive an appropriate explanation of the data. Our belief is that establishing of data-driven approaches like these leads to more generalizable techniques. We proceed with the understanding that customization of a general approach for specific constrained situations should produce equal or better results.

Allowing the *structure* to be data-driven permits modeling of new articulated subjects without introducing user error or range-of-freedom limiting approximations. Data-driven *correspondence* serves the same purpose as tracking, namely to automate acquisition of data sequences. Finally, the *parameterization* integrates structure and correspondence, making them useful for analysis and synthesis.

This research makes contributions to the ability of a machine, using computer vision, to perform data-driven analysis of articulated movements, exhibited primarily by human and animal subjects. We explore the requisite algorithms for an automatic image-based technique that determines a creature's *Spine*: an intermediate parameterized model of both articulation and surface deformation. The Spine is a chain of oriented 3D nodes that snakes through the middle of each body part and branches at limb-junctions. In general, our approach combines the objectives of generalized cylinders, 3D scanning, and markerless motion capture to generate baseline models from real puppets, animals, and human subjects.

Our method recovers this parameterization by combining the visual constraints imposed by videos of a performance. Synchronous video footage from multiple angles is merged to produce a sequence of 3D volume representations. Each volume in the sequence constrains the subject's possible pose at that time. Our technique for parameterizing this volume data automatically constructs a single computer graphics model of the subject, complete with limb-points and correspondences that guide subsequent tracking. Our thesis states:

> *A functional structure and the time-varying surface of an articulated creature subject are contained in a sequence of its 3D data.*

The functional structure is one example of the possible arrangements of internal mechanisms (kinematic joints, springs, *etc.*) that is capable of performing the motions observed in the 3D input data. Among such possible mechanisms, the notion of functional structure also includes the real osteology of the creature, that we are not seeking. For both tracking and synthesis purposes, the construction of an internal structure is usually accomplished

manually by skilled modelers who "rig" the character's (or intended tracking-subject's) bones and skin. There is no single rigging that would satisfy all applications. Therefore, as a foundation, we present a general and repeatable model of a creature's limbs and their correspondence over time and pose, as applies to creatures that demonstrate their articulations while moving.

After the following chapter on related work, Chapter 3 describes the theory and process of Spine extraction. We explain the stages of our algorithm for converting multi-view video into a sequence of meshes, then a sequence of estimated skeletons, and finally a single time-varying Spine. This chapter also includes the development of a new algorithm for merging tree-graphs using leaf-node correspondences. Further, we detail the process of acquiring experimental data for several creatures (see Figure 1).

There is only limited real volume data of moving and articulated subjects, so direct evaluation of the surfaces we extract is difficult. However, the technology of motion capture (mocap) has developed reliable means for tracking individual 3D features, and Chapter 5 discusses how to quantitatively compare our *Spine* data with mocap data. We also address the respective advantages of animating using the two different techniques.

Functional evaluation of the structural Spines we generate is more challenging, because the generalizability of such Spines as armatures is application-dependant. These dependencies leave many areas open for further research. We conclude the dissertation with a general discussion.

1.1 Broader Impact and Applications

The study of natural phenomena is based on both improved analysis of existing data, and technical advances in our ability to capture new data. While human and animal motion had been studied long before the photography of Etienne-Jules Marey [14] or Eadweard Muybridge [61], their technological advances were used to capture data that resolved, among other things, long-standing questions about movements and specifically about gait.

Figure 1: Photos of five subjects filmed to obtain experimental data. The middle image shows the surface of the adult human, and pictured clockwise from the upper left is the human baby (11 months old), the Rosehair tarantula, the dog, and the camel marionette.

This trend continues with modern motion capture systems [72]. The same is true of lighting with comparatively recent work on capturing of Light Fields/Lumigraph rendering [45, 33]. Even newer work performs analysis and synthesis of lighting on materials and textures [50, 43, 26]. In addition, the technological developments most relevant to our work deal with surface acquisition or range-scanning. The same data once captured only by talented sculptors like Michelangelo, can now be scanned [23, 1], as can increasingly (see NSF Grant #0121239 [2]) the historical and archeological artifacts themselves [38, 46, 6].

These technological advances in particular have presented us with new forms of data that capture and represent relevant detail. Here we examine a new data representation that combines the advances in motion capture and surface acquisition. In this document, we

propose an approach for capturing 3D surface and structural data of moving articulated subjects. The increase in number of research groups dealing with sequences of voxel and polygon data from multi-view cameras ([70, 19, 53, 12, 20]) is indicative of the need for a data-driven representation stripped as much as possible of heuristics and prior knowledge.

Furthermore, motion data is the established source of our knowledge about biomechanical characteristics of humans and animals. Historical developments in capture technology have led us from multi-exposure photography through motion pictures to modern motion capture. The quality of the data depends not only on its accuracy, but also on our ability to analyze the results – manually or automatically. Significant portions of the computer vision field analyze the tracking of pose and surface deformation data, though the two are usually treated separately. Instead, joint analysis of moving surfaces and their underlying structure can be pursued thanks to the parallel improvements in multi-view video technology and the new capabilities in geometric analysis of shapes. The underlying structure is of particular interest.

Both character animation and articulated-motion tracking benefit from using an underlying kinematic model of the subject. For animation, such a model, or armature, exposes only the handles required to control the important degrees of freedom, much like the strings on a puppet. For tracking, a model constrains the problem to a search for a comparatively limited number of pose-parameters. Armatures designed for either task have the same tradeoffs; simple models have few degrees of freedom (DOFs), making them easier to manipulate, but complicated models are capable of embodying nature's structures more precisely.

This thesis is a partial response to what we subjectively perceive as heuristics-based armature-building. Until now, for lack of a better solution, most applications start with a hand built and often expressly initialized armature. Consequently, when evaluating the performance of a tracking system, it is non-trivial to distinguish errors inherent to the tracking

from those caused by disparity between the subject and the "representative" model. Imperfections in an animation armature are even harder to detect, because the practice of adding surface deformation DOFs is accepted for purposes of expressiveness, and imperfections are whittled away by artists' iterative adjustments. In the following chapters we intend to show that subject-specific armature information is implicit in sequential volume data, without need of heuristics.

1.2 Mathematical Objective

We are interested in the detection and tracking of features in volumetric images. Volume images capture shape as a temporal sequence of boundary voxels or other forms of 3D surfaces. Specifically, we wish to address situations where the subject is known to have and is exercising an articulated structure. This assumption grants us use of a specific class of geometric modeling solutions. The various methods for skeletonizing 2D and 3D images share the objectives of identifying extrema, features with some geometric significance, and capturing the spatial relationships between them [20]. Skeletons, much like generalized cylinders [9, 51], serve the purpose of abstracting from raw volume or surface data to get higher level structural information.

We propose that evaluating volumetric data of a subject over *time* can disambiguate real limbs from noisy protrusions. In a single image, knowledge of the specific application alone would dictate the noise threshold to keep or cull small branches of the skeleton. Many such algorithms exist. In the case of articulated moving subjects, the volumetric images change but the underlying structure stays the same. We propose that the parts of the skeleton within each image that are consistent over time more reliably capture the subject's structure. To this end, we introduce our notion of Spines.

As defined by Binford [9], a generalized cylinder is a surface obtained by sweeping a planar cross section along an axis, or space curve. To represent a body made of multiple generalized cylinders, we need to merge axes of the different limbs into one branching axial

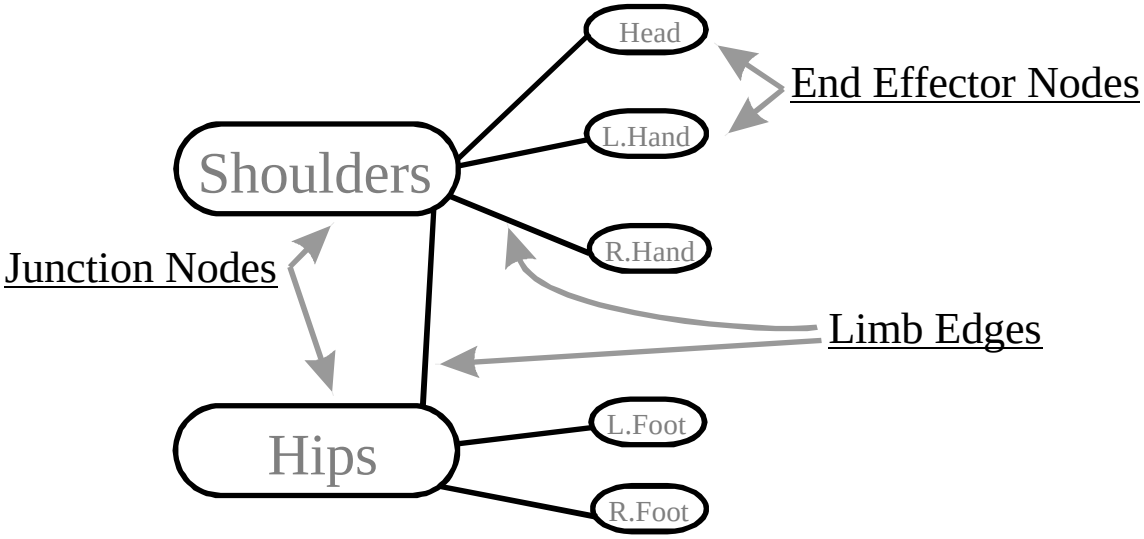


Figure 2: Spine graph limbs encoding motion over time; nodes labeled for illustration only.

structure. The branching structure can be represented by a graph, $G(LimbBoundaries, Limbs)$, where edges are limbs, leaf nodes are end effectors, and the remaining nodes (all of degree > 2) are limb junctions (see Figure 2). So far, we have described the general formulation of a skeleton [11]. To parameterize the motion of a skeleton, we express the new Spine graph as a function over time:

$$Spine_t = F(G,t). \tag{1}$$

For a given time t, the limbs of G will be in a specific pose, captured by F's mapping of G's topology to axial curves in 3D – a single *skeleton*. When estimating a data set's *Spine* in the subsequent sections, we will constrain F to manipulate the limbs of a G that represents a series of topologically consistent skeletons. These skeletons are determined as probable given the input data.

The implementation of our algorithm is a modular pipeline. It first reduces the complexity of multi-view video data to voxels, further to polygons, and finally to Spines. The resulting model captures the original degrees of freedom needed to play back the subject's

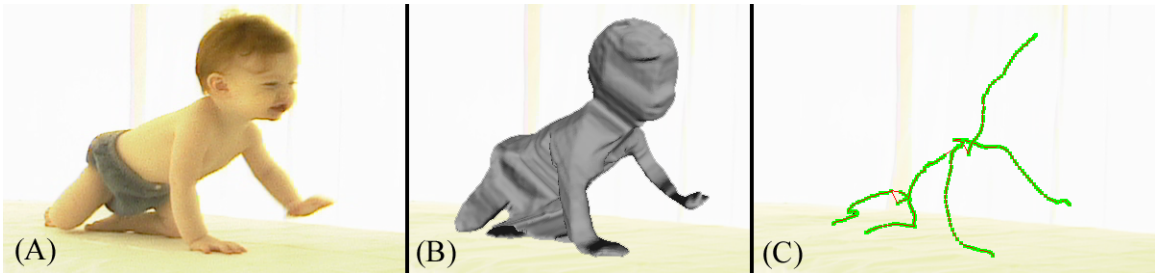


Figure 3: (A) Articulated subject, (B) reconstructed surface, (C) extracted skeleton.

motions (see Figure 3).

CHAPTER II

RELATED WORK

This work builds on progress made in the areas of image-based modeling and mesh skeletonization. The progress recently made in markerless video-tracking, and deformation by example serves as motivation for our approach. We briefly discuss all these here.

Image-based Modeling: With increasing effectiveness, multi-camera environments with intersecting view volumes are being used to reconstruct 3D surfaces. The initial voxel carving work used just the subject's silhouettes in each image to carve away empty parts of the volume. The works of Kutulakos & Seitz and Seitz & Dyer [44, 65] developed the notion of Photo Hulls and used the additional information provided by pixel colors to model visible concavities [22].

The 50 plus camera system of Vedula et al. [70] has been used to record sequences of volumes, that are later converted from voxels to painted polygons. The intended application here and in the faster polygon-based work of Matusik et al. [53] is the playback of virtual versions of motion sequences. Thus far, these techniques have focused on the visual realism of the resulting surface representation. Our contribution to this field is the analysis and modeling of the interior structure of the volume over time.

Table 1: Context of our approach (marked in color) with respect to existing algorithms in established fields.

Feature Tracking	Surface Acquisition	Shape Modeling
2D - points / corners 3D - markers	Structured Light Laser Range Scanning Stereo	Medial Axis Transform Principal Curves
4D - limbs / extrema	Silhouette/Generalized Voxel Carving	Geodesic Level Sets

Medial Axes and Mesh Skeletonization: The 2D analogue to our problem is the tracking of correspondence in medial axes, that were first introduced by Blum [11]. Given any of the numerous 2D skeletonizing techniques, including the classic grassfire models based on distance and the more robust area-based techniques [8], the work of Sebastian et al. [64] can determine correspondence by minimizing edit-distances of skeleton graphs in 2D.

The medial axes of 3D surfaces are not directly applicable because they generate 2D manifold "sheets" through a surface. While medial scaffolds can be calculated fairly robustly [66, 48], they require further processing [73] to estimate good 1D axes.

Several 3D skeletonization algorithms have been developed using 3D Voronoi cells to partition the space within a mesh [5, 30, 69, 28, 36]. The cell-walls of these convex polyhedra land at equal distances from their designated surface start-points – some at or near the medial axis. This approach, with various extensions of projection and pruning, can generally serve to synthesize axes. In this way, Hubbard [36] generated a center-spine for meshes whose collision detection is accelerated through the use of bounding spheres. Similarly, Mortara & Spagnuolo [56] compare their Delauney-based Approximate Skeleton to real 2D medial axes for purposes of object-correspondence and morphing. As was shown by Teichmann & Teller [69], these approximate skeletons are very sensitive to surface variations but, with care, can be cleaned up by a user wishing to build a character-animation armature. In contrast to these, our approach and implementation are based on two sub-domains of solutions: measuring of geodesic distance from geometric modeling, and principal curves from statistics.

The popularity of meshes and their inherent lack of internal structure have led to methods for extracting this structure. Hilaga et al. [35] developed a surface isolation metric for locating extremities to measure similarities between 3D shapes. That metric starts to address the significant computational cost of performing 3D object-recognition like that done in matching shock-graphs in 2D by Sebastian et al. [64] or matching 1D strings.

Except for Siddiqi et al. [66] who obtain real medial-axis surfaces from medical volume

data of rigid objects, the rest of those mentioned here seek a stick-figure form of the input geometry. Li et al. [49] is one recent example of a class of papers that build skeletal edges by mesh simplification. In this case they progressively contract the longest edge until the rough structure of the mesh results. However, we found this approach erodes appendages and frequently places skeleton nodes outside the original geometry.

Geodesic Distance: In Section 3.2 we will discuss in greater detail how a surface can be treated as a piecewise continuous distance field that separates features from each other. Verroust and Lazarus [71] used such a technique to determine axes of symmetry within limbs, and how to connect them to critical points (special topological features) on the mesh surface. In an application not requiring branching axes, Nain et al. [57] used geodesic distances on colon models to determine center-lines for virtual colonoscopy navigation. Recently, a geodesic distance based metric was used by Katz and Tal [41] to help assign patches as members of explicit limbs, resulting in coarse animation control-skeletons. All these approaches benefit from works such as Hilaga et al. [35] that identify extrema, or features that protrude from or into a surface mesh. Our approach uses such extrema-finding and a geodesic distance metric to better model skeleton branching.

Principal Curves: Hastie and Stuetzle [34] defined principal curves as passing through the middle of a multidimensional data set, as a representation of self-consistency to generalize principal components. For fixed length curves in a geometric setting, Kegl et al. [42] showed how to minimize the squared distance between the curve and points sampled randomly from the encompassing shape. Most recently, Cao [16] and Cao & Mumford [17] extended this notion of principal curves to 3D, formalizing the problem as an optimization that also seeks to minimize the curve length. Our extension is to incorporate branching and temporal correspondence.

Markerless Motion Capture: As we are interested in tracking articulated structures over time using video, it is important to also consider the state of the art in vision-based motion

tracking and capture of people. Most methods in this space use appearance, templates, or feature-based tracking to track hand initialized limbs [67, 40]. In most situations an *a priori* model of the structure is provided to aid in tracking and reduce complexity [27, 25, 59, 15]. Plänkers & P. Fua [60] use silhouette and stereo data of people to fit Metaballs using a least-squares approach to generate articulated models of humans. Only a few efforts have explored multi-view analysis [15, 32]. Mikic et al. [55] extend the appearance-based approaches to multiple views and use voxels to fit cylinders in place of affine patches. The major difference in our approach is that we do not rely on any predefined model of articulation and we use the data to generate the underlying skeleton. This enhancement allows us to capture motion data of any type of articulated subject.

Deformation By Example Interesting techniques are being developed to create relationships between a character's pose and the deformations on their surface. Fundamentally, this task is a problem of interpolating between deformed example-surfaces that are known to be good [47, 68]. The underlying skeleton helps drive the parameterized interpolation along realistic trajectories. However, even with these techniques, insufficient examples allow the radial basis functions to short-circuit real skin-trajectories, allowing a character's skin to crumple and self-intersect.

Most example deformations are specified by an artist modifying interpolation weights on a single template mesh as given poses need repairs. The recent work of Allen et al. [1] shows that, with some effort, these examples can be captured from the real world via a range-scanner. After the user constructs an approximate ball and socket armature, builds a subdivision surface for the intended body parts, and labels the body-markers, the approximately 100 range-scans refine the model's kinematics and surface to match the subject. Our approach captures whole sequences of naturally occurring deformation examples, and does not require manual model-building.

CHAPTER III

SPINE FORMULATION & ESTIMATION

In this chapter, we build on the axial representation of generalized cylinders of Cao & Mumford [17, 16] because of their elegant mathematical formulation. They treat the regression problem of finding a single curve for a surface as the minimization of a global energy function. Much like the previous work on principal curves [34, 42], they seek to minimize the total distance from the axial curve to the surface. But in addition, Cao [16] incorporates a term that penalizes the length of the curve. This augmentation helps force the shorter curve to smoothly follow the middle of a surface, instead of, for example, spiraling through all the boundary points.

3.1 Formulation

For our Spine formulation, we seek to further incorporate: (a) skeletons S that model branching curves of individual surfaces X and (b) data captured over a period of time T. We propose a discriminative probabilistic approach to computing Spines by finding G, S, and limb end effectors E, that maximize:

$$P(G, S_{1:T}, E_{1:T} | X_{1:T}) = P(G | S_{1:T}, E_{1:T}, X_{1:T}) \cdot P(S_{1:T}, E_{1:T} | X_{1:T}) \tag{2}$$

To compute and optimize the joint probability $P(S_{1:T}, E_{1:T} | X_{1:T})$ requires searching over all skeletons over all time simultaneously. In order to make the solution more computationally tractable, we make the assumption that S_t and E_t are independent of $S_{t'}$ and $E_{t'}$ $\forall (t' \neq t)$, given X_t:

$$P(G, S_{1:T}, E_{1:T} | X_{1:T}) \approx P(G | S_{1:T}, E_{1:T}, X_{1:T}) \cdot \prod_{t=1}^{T} P(S_t, E_t | X_t) \tag{3}$$

This assumption can lead to temporal inconsistencies that can be resolved once G is estimated (as shown in Section 3.3). We use a bottom-up approach that individually approximates each S_t and E_t individually, and then estimates G. Ideally, we would like to estimate G, S, and E using an EM-like algorithm by iterating back and forth between estimates of G and (S_t, E_t). However, we have found that the greedy estimate of S and E, while noisy, is usually sufficient to determine a G consistent with the subject's limb topology – to the extent that the motion explores relevant degrees of freedom.

In this section, we will start by describing our method for locating the set of end effectors E_t and extracting a branching skeleton graph from a single 3D surface X_t. Using this or other techniques, we can generate an individual skeleton S_t at each time t, $1 \leq t \leq T$. These (S_t, E_t) will be inherently noisy, as a result of being calculated independently for each t. In Section 3.3, we describe how we combine these individual and often overly complex graphs into a consistent, representative Spine for the entire time sequence.

The fairly significant attention given to the problem of building a single branching 3D skeleton includes numerous approaches (see the Mesh Skeletonization section in Chapter 2. After experimenting with portions of several of these [49, 35], we have developed our own extension to the level-set method of Verroust & Lazarus [71]. In theory, any 3D skeleton-finding technique would be suitable, if it meets the following requirements:

1. Is self-initializing by automatically finding extrema E_t.

2. Generates a principal curve leading to each extremum.

3. Constructs internal junctions of curves only as necessary to make a connected tree.

More precision might be achieved with more iterations or other techniques, but these might only further improve the results of applying our general probabilistic framework of Equation 3. We proceed to explain our greedy method for obtaining a 3D branching skeleton S_t from a surface, with just one iteration of maximizing the second term of Equation 3, followed by correspondence tracking.

3.2 Creating a Spine for a Single Frame

After obtaining a 3D surface for a frame, we want to extract a skeleton from it. Ideally, one would like for the skeleton to trace out the middle of the 3D surface, extending to the tips of the subject's extremities. For example, the skeleton of a starfish should be five branches that extend radially from the center through the middle of each arm.

Once we have a 3D surface X_t for volumetric image (or frame) t, we want to extract a skeleton from it. We accomplish this goal in two stages. First we find the tip of each extremity and grow a skeleton from it. Then we merge the resulting skeletons to maximize the presence of the highest quality portions of each. In terms of maximizing $P(S_t, E_t|X_t)$, we first find a set of candidates for the end effectors of E_t and the limbs of S_t. We then pick from these the limbs that are optimal with respect to our probability metric.

Growing Skeletons: This part of our algorithm is based on the work of Verroust & Lazarus [71]. Starting at a seed point on an extremity of the mesh, they sweep through the surface vertices, labelling each with its increasing geodesic distance. These distances are treated as a gradient vector field, that is in turn examined for topological critical points. The critical points are used as surface attachment sites for virtual links (non-centered) between the axes when the mesh branches.

But for our purposes, we want a skeleton that always traverses through the middle of the subject's extremities. Locating meaningful extremal points is itself an open problem, though the difficulties are generally application specific. Much like the above algorithm that has one source, the vertices of a surface mesh can be labelled with their *average* geodesic distance (AGD) to *all* other points. Surface points thus evaluated to be local extrema of the AGD function correspond to protrusions. Knowledge of the expected size of "interesting" protrusions can be used as a threshold on which local maxima qualify as global extrema.

Hilaga et al. [35] address the significant computational cost of finding the AGD by approximating it with uniformly distributed base seed-points. Applying the simpler base-point initialization of Verroust & Lazarus and Cormen et al. [71, 21] in a greedy manner,

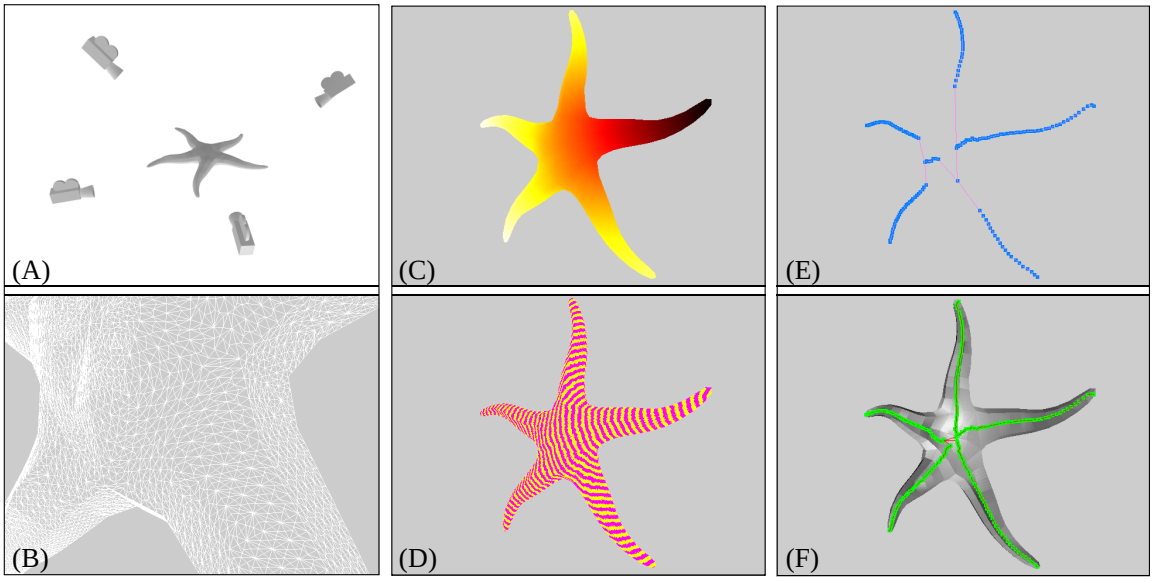


Figure 4: Example of generating a skeleton for a synthetic starfish mesh. (A) Capture images of the starfish from a variety of vantage points (B) Extract a 3D surface using generalized voxel carving and improved marching cubes (C) Starting at one extremity tip, calculate geodesic distances for each vertex (D) Quantize distances and cluster vertices into bins of the same distance (E) Create a skeleton by walking through the progression of level set rings (F) Repeat C-E for each tip and merge into a single representative skeleton.

located the desired candidates for E_t for our data sets.

Instead of the separate *distance* and *length* terms minimized by Cao [16], we use the isocontours of geodesic distance to build level sets that serve as our error metric. The vertices of the mesh are clustered into those level-sets by quantizing their distances from the seed point into a fixed number of discrete bins (usually 100). Figures 4C-D illustrate this process. Each skeleton node is constructed by minimizing the distance between the vertices in the level set and the node, *i.e.*, the centroid of the vertices.

By walking along edges of the surface graph from the seed point's level set toward the last one, skeleton-nodes are added and progressively connected to each other. Figure 5 illustrates this process in 2D. This approach successfully creates a tree graph of nodes, or skeleton, that represents the central axes and internal branching points of genus zero meshes.

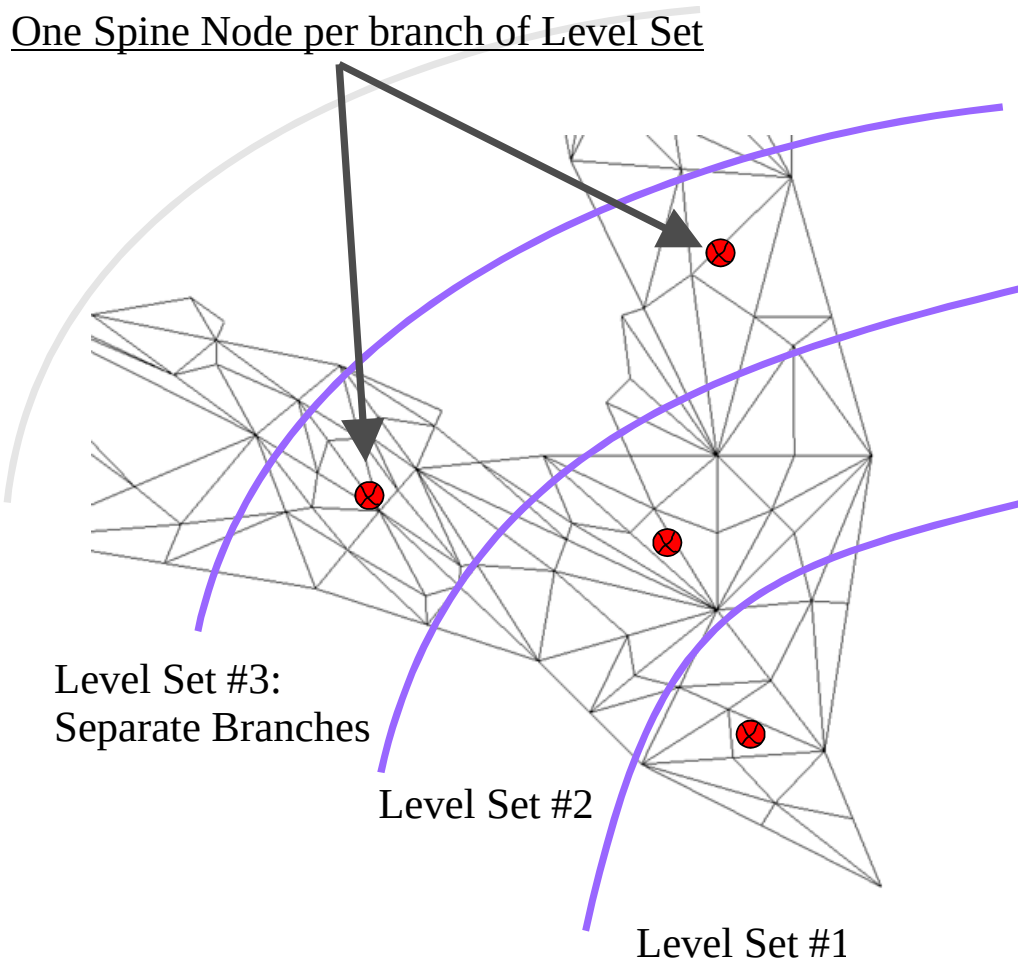


Figure 5: 2D example of clustering connected vertices into bins of similar geodesic distance and walking through the resulting level set rings.

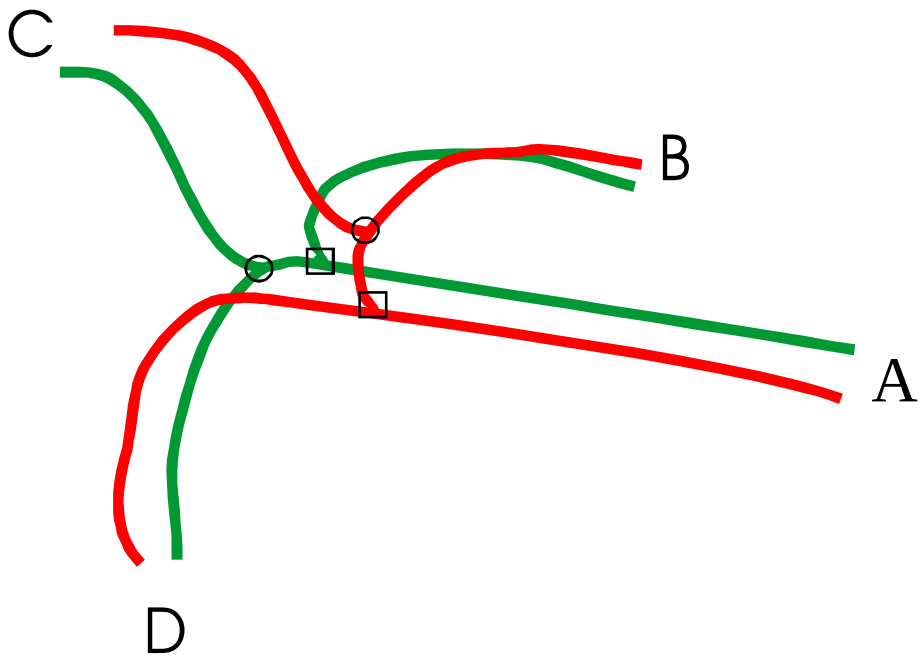


Figure 6: The red and green skeletons represent the same "creature," possibly seeded from two different places. Wishing to copy nodes from the best limbs each constituent skeleton has to offer, we developed a leaf-node seeking topology matching algorithm that recognizes that these pairs of three-way junctions should be a single four-way junction.

The skeleton-generation algorithm is repeated for each of the other limb-tips, producing a total of five skeleton-graphs for the starfish example (see Figure 4). These are our candidates for the best S_t for this X_t. Note that the most compact level-sets usually appear as tidy cylindrical rings on the limb where that respective skeleton was seeded.

Merging Skeletons: All of the constituent skeletons S_t serve as combined estimates of the mesh's underlying limb structure. The best representation of that structure comes from unifying the most precise branches of those skeletons – the ones with smallest error, or equivalently, maximum $P(S_t, E_t|X_t)$. A high quality skeleton node best captures the shape of its "ring" of vertices when the ring is short and has small major and minor axes. With this metric, we calculate a cost function C for each node in the constituent skeletons:

$$C_i = \frac{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}{\#\text{ of points in ring } i}. \tag{4}$$

The σ quantities come from singular values of the decomposition $\bar{\mathbf{P}} = \mathbf{U}_P \Sigma_P \mathbf{V}_P^T$, where $\bar{\mathbf{P}}$ represents the mean-centered coordinates of the points p_i in this ring.

Note that the resulting $\mathbf{v}_i$ vectors in $\mathbf{V}_P^T = \{\mathbf{v}_1 | \mathbf{v}_2 | \mathbf{v}_3\}^T$ will usually represent the major, minor, and central axes of the ring. Replacing $\mathbf{v}_3$ with $\mathbf{v}_1 \times \mathbf{v}_2$ produces a convenient local right-hand coordinate frame for each node.

Each chain of bi-connected nodes represents a limb. To assemble the single representative graph of this frame, we copy the best version of each limb available in the constituent skeletons. Limb quality $\mathbf{Q}_L$ is measured as

$$\mathbf{Q}_L = N - \sum_{1}^{N} C_i, \tag{5}$$

where N is the total number of nodes in limb L. As nodes from different skeletons are being compared through Equation 5, the cost of each node must be normalized by dividing them all by the $\max(C_i)$ of all the skeletons.

Figure 6 illustrates a novel algorithm that we developed to generate limb-correspondences for topologically perturbed tree graphs of the same structure. There appears to be no previously established graph theoretic solution for this problem, and our approach is simply

1. Tag all limb-tips that we are confident of as *Supernodes*; i.e. nodes on both color graphs located at [A, B, C, D] correspond to each other.

2. Traversing inward, the next encountered branch-node in each graph also corresponds to that of the other color: walking from supernode A, the skeleton-nodes at the square-symbols should be grouped into a supernode of their own. From C, the circles will form a supernode. Iterating this process from the outside inward will reveal that the circle and square supernodes should be merged into a four-way *metanode*, that would serve as the point of unification when merging limbs from the red and green skeletons.

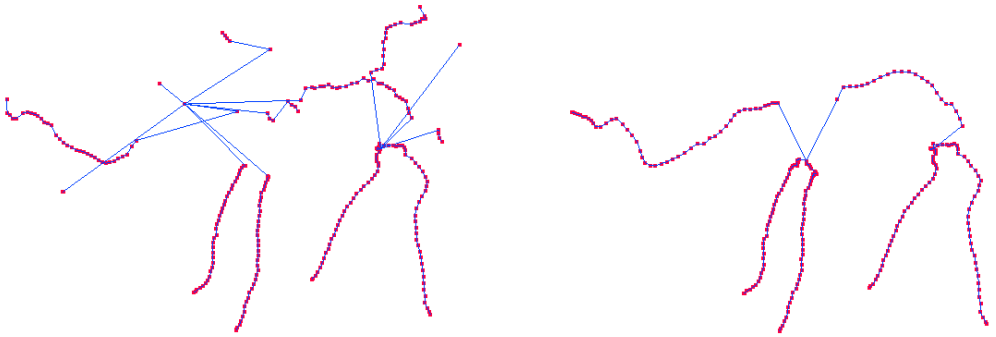
3.3 Correspondence Tracking

Now that we can estimate a single skeleton that represents one volumetric image, we adapt the process to handle a sequence of volumes. All the measurements from the sequence of $X_{1:T}$ are now abstracted as $(S_{1:T}, E_{1:T})$, simplifying the first term in Equation 3 to $P(G|S_{1:T}, E_{1:T})$. Finding the G that maximizes this probability eliminates extraneous limbs that might have resulted from overfitting. The danger of overfitting exists because skeleton elements may be created in support of surface-mesh elements that looked like protrusions in that frame only.

Our 3D correspondence problem of finding the best G is significantly easier to automate than trying to perform surface-vertex matching between two dense meshes of the sequence. Assuming the subject grows no new appendages and with no other priors, we can choose the appropriate number of tips to be the most frequently observed number of limb tips. This number of tips, or leaf nodes in G, is $K =$ the mode of $|E_t|$, $1 \leq t \leq T$ (see Figure 16).

Knowing how many appendages to look for, we spatially align each exploratory skeleton from the sequence with respect to its temporal neighbors to reveal the $|E_t| - K$ superfluous tips that should be culled. We start with all the subsequences of frames that already have the correct number of tips K, and tag the frame from the middle of the largest such cluster as the reference frame; allowing that longer sequences may need to automatically select multiple reference frames. Each frame is then processed in turn, constructing a combinatorial list of possible tip-correspondences between the reference tips $\mathbf{A}$ and the tips in the current frame $\mathbf{B}$. Each possible mapping of $\mathbf{B} \rightarrow \mathbf{A}$ is evaluated using the point-cluster alignment algorithm of Arun et al. [3]. Their technique aligns point clouds as much as possible using only translation and rotation. The combination with the smallest error, $E_{\min}$, is kept as the correct assignment, where

$$E = \sum_{k=1}^{K} \|B_k - \hat{\mathbf{R}} A_k - \hat{\mathbf{T}}\|^2. \tag{6}$$



A: Without refinement B: With temporal constraint

Figure 7: Refinement through imposing of correspondence into the sequence. Instead of greedily including every protrusion that appears to be an end effector, we are able to keep only the limbs that appear consistently over time.

Here $\hat{\mathbf{R}}$ and $\hat{\mathbf{T}}$ are the least-squares optimal rotation and translation. $\hat{\mathbf{T}}$ simply comes from aligning centroids of the point clouds. $\hat{\mathbf{R}}$ is calculated by maximizing the $Trace(\hat{\mathbf{R}}\mathbf{H})$, where $\mathbf{H}$ is the accumulated point correlation matrix:

$$\mathbf{H} = \sum_{k=1}^{K} A_k B_k^T. \tag{7}$$

By decomposing $\mathbf{H} = \mathbf{U}_R \Sigma_R \mathbf{V}_R^T$, the optimal rotation is

$$\hat{\mathbf{R}} = \mathbf{V}_R \mathbf{U}_R^T. \tag{8}$$

After assigning the tips of all these frames, we apply the same error metric to try out the combinations of tip-assignments with frames having alternate numbers of tips. However, these frames are compared to both the reference frame and the frame nearest in time with K tips. This brute-force exploration of correspondence is computationally tractable and robust for creatures that exhibit some asymmetry and have a reasonable number of limbs (typically < 10).

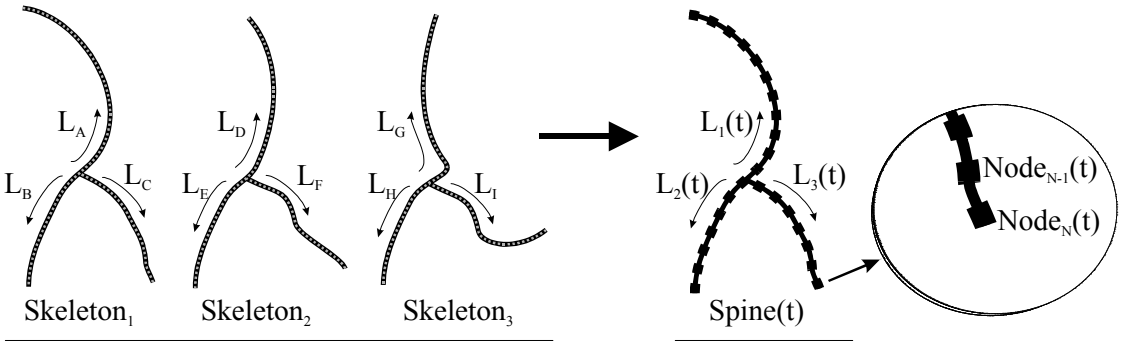


Figure 8: The sequence of skeleton-trees (left) has separate node-branches $L_A..L_I$. The limb-to-limb correspondence is known across time, but each node exists only in one limb for one frame. Normalizing each limb's length with respect to time, we resample the Spine to form one set of Spine-nodes (right) whose position varies as a function of time.

3.4 Imposing a Single Graph on the Spine

With the known trajectories of corresponding limb tips throughout the sequence, we can re-apply the skeleton merging technique from Section 3.2. This time however, we do not keep all the limbs as we did in the exploratory phase, only those that correspond to the *K* limb-tips. The results of this portion of the algorithm are pictured in Figure 7 and discussed further in Section 4.3.

Except for the frames of the sequence where the subject's limbs were hidden or tucked too close to the body, we can expect the topology of skeletons throughout the sequence to be identical. The most frequently occurring topology is established as *G*, and corresponds to the first term in Equation 3. This correspondence and trajectory information allows us to construct a single character Spine for playback of the whole sequence of poses by parameterizing on each limb's length. Each topologically consistent limb of the skeleton sequence is resampled at the same interval producing a single Spine. Figure 8 illustrates how a sequence of skeletons, once aligned with known correspondence, can have their limbs resampled. The elements of the resulting Spine (nodes and limbs) can now be indexed or interpolated according to time.

CHAPTER IV

IMPLEMENTATION & EXPERIMENTAL RESULTS

4.1 Design Decisions

To test our approach, we required a sequence of volumes or surfaces capturing the motion of various articulated creatures, preferably without markers or other props. This type of data has been rare, and was difficult to obtain because most imaging is done with only one or two cameras, and volumetric reconstructions are most common in biomedical applications, where internal organs are being scanned. Further, existing 3D scans of humans and animals have tended to be captured with laser range scanning or other techniques that require that the subject stand still for several seconds. Seeking a large range of realistic poses, we needed to record full-body volumetric data of our subjects at sampling rates corresponding to the speeds of their motions. We have found consumer grade video framerates of 30 frames per second sufficient for many of our subjects, allowing that faster motions would require still faster sampling.

Forced to obtain our own data, we developed a video based surface reconstruction pipeline, with Generalized Voxel Carving (GVC) [22] at its heart. Various parts of the pipeline could be replaced by other techniques, including the GVC stage itself, but new techniques should only improve the final sequence of surface reconstructions, and consequently our Spine estimation results. In Section 4.2, we elaborate on the details of the individual reconstruction stages, along with explanations for the specific design decisions.

All our data of moving subjects was acquired using the process described in Section 4.2, with the exception of the Adult Human polygonal mesh sequence, that was kindly provided by Trevor Darrell's group [24] at the MIT CSAI Lab. Of these, only the tarantula was filmed with upward pointing cameras (transparent flooring), because other subjects were too heavy

for our plexiglass. All possible cameras were used for each reconstruction, but a subset had to be excluded for varying reasons; some cameras were obstructed because subjects required human proximity, others were left out because of difficulty with calibration.

4.2 Data Acquisition and Processing

We applied the algorithm on a variety of small creatures after building a data capture stage that would both be comfortable for the subjects and minimize the need for video segmentation beyond chromakeying. Twenty video cameras were attached to an aluminum exoskeleton shaped roughly like a cylinder 3 meters in diameter. Their viewing angles were chosen heuristically to maximize viewing coverage of subjects on a raised platform, and to minimize instances of cameras seeing each other's lenses. The capture volume itself is $(75cm)^3$, and can accommodate creatures that stay within the space (Figure 10). While the following pipeline was designed to preprocess data and then handle the results of GVC, Visual Hull and dense stereo reconstruction implementations would benefit from similar designs.

Capture Stage Construction: The capture stage serves the dual purposes of providing a structure for attaching cameras, and to simplify the subsequent task of video segmentation by providing a uniform background. To allow the cameras to be mounted in a fashion encircling the subject and pointing inward, we used an exoskeleton design as pictured in Figure 9. The finished capture stage and lighting are pictured in Figure 10. The structure is built of aluminum because of its light weight, but the plastic junction pieces left the overall structure lacking in rigidity. As cross-beams would have obstructed the views of some cameras, the top corners of the hexagonal structure were anchored with tension ropes to weights on the outskirts of the room. An all wood exoskeleton might have saved this step, but the thicker beams would have blocked more light.

The material covering the walls of the enclosure was selected based on its ability to

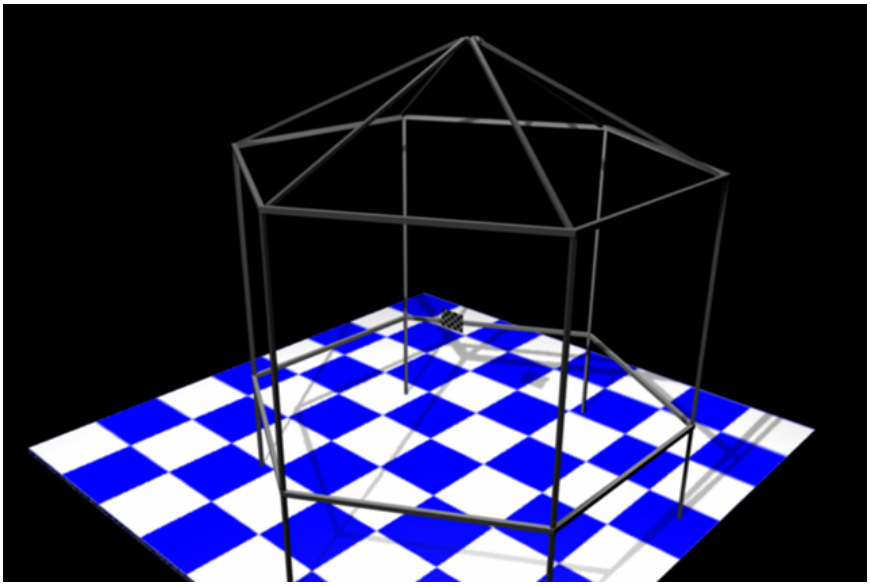


Figure 9: Construction previsualization of our data capture stage. The structure is designed to accommodate small subjects, and support 20 or more cameras without obstructing incoming light. Note “floating” calibration pattern of dimensions 20 x 14cm.

transmit light, yet scatter shadows and colors coming from the outside, making the interior appear uniform. Shadows of the exoskeleton and cameras pressed against the sides were of particular concern. After experiments with various types of cloth and material, we chose white shower curtains, whose color offered the advantage that light shining through would not alter the colors of the subject. The large shower curtains were glued together into a four meter tall tarp that was suspended from the hexagonal structure, approximating the shape of a cylinder with a rounded bottom.

We experimented with several types of lighting, and settled on fluorescent light fixtures. They have the advantage of appearing the most like area lights when shining through the shower curtain material. By distributing the lights around the room and placing them, for the most part, between cameras, we were able to minimize the casting of shadows onto the material. This arrangement proved very effective at making the subject’s appearance contrast with the background, and essentially appear to be illuminated only by ambient light.

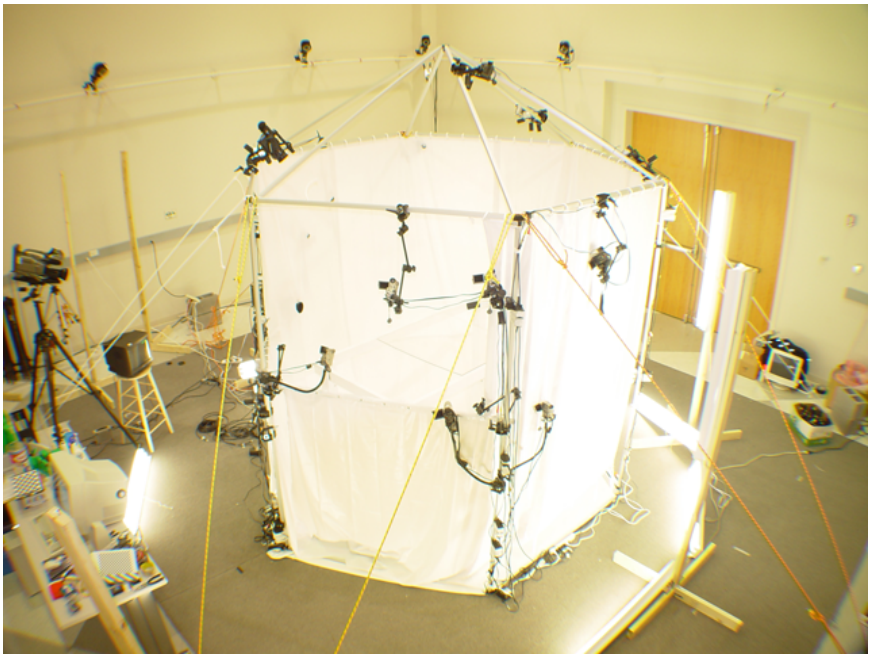


Figure 10: Data capture stage surrounding a platform with an interchangeable table top.

Cameras and Calibration: Ideally we would have used progressive scan cameras that accepted an external synchronization signal and ran at full video frame rate (30 frames per second) for several minutes. However, we instead had access to a collection of progressive scan DV cameras, the Canon Elura® 1, 2, and 20 series, that have built-in recording to tape. These have good optics and allow all of the controls to be set manually, but lacked the ability to synchronize, as they are mostly meant for consumer use. Satisfactory synchronization was possible by turning on their power sources at the same instant using surge protectors. Care was taken to adjust their exposure and white balance settings to be consistent, so that colors were seen consistently from all sides.

Intrinsic calibration of the cameras was achieved using the camera calibration toolkit distributed with Intel's OpenCV library [13]. This calibration provided us with the x and y focal lengths, principal point, and coefficients on radial lens distortion for each camera. For some of the subjects, the extrinsic parameters, camera center position and orientation, were obtained using the same toolkit. One pose of the checkerboard was picked as the home

coordinate frame, and extrinsics for cameras that could not see it clearly were transformed by chaining transforms of other checkerboard orientations. The extrinsic calibration technique used for the Baby and Dog data was based on a Levenberg-Marquardt optimization of point locations of a wand that was waved in front of all the cameras at once.

Background Segmentation: Along with filming of principal footage of each subject, we also kept footage when the subject was outside the field of view. These "empty" sequences were median filtered in time to produce a representative background image for each camera. While the subjects generally stood out as contrasting with the color of the background, the background was rarely a single uniform color, requiring more than just chromakeying. Our system for background subtraction used a combination of color difference and blob size to isolate the foreground elements. With only a few exceptions, a color difference threshold of $20/255$ gave acceptable segmentation for whole sequences of subjects filmed in our environment. Problems arose when the subject either reflected in the table, or cast a large shadow similar in color to its own skin. Segmentation continues to be an important area of research, and our implementation could certainly benefit from further innovations. The brightness of some footage had to be boosted when subjects appeared at times to be completely black (*i.e.*, having RGB$= 0,0,0$). That color is reserved in the voxel carving stage to indicate background pixels.

Specifying Bounding Volumes: Using the camera calibration information for each camera allows us to reduce the size of the bounding volume, reducing the number of superfluous empty voxels carved. Either all or at least two camera's videos of the subject are processed to find the respective sequences of 2D bounding polygons. These are passed to our system for producing a single 3D bounding box (origin and $height * width * depth$ dimensions) for each sample t, $1 \leq t \leq T$. The system projects the sides of a large bounding box on top of each of the 2D bounding polygons according to the camera calibrations. It performs a binary search in shrinking the dimensions of the 3D bounding box until the box just encloses

all the 2D bounding polygons.

Volumetric and Surface Reconstruction: Greg Slabaugh and his colleagues at Hewlett-Packard Laboratories were kind enough to provide us with an implementation of their Generalized Voxel Carving (GVC) system [22]. After preprocessing the data as described above, we simply selected the desired real world dimension of each voxel before running GVC on the calibrations and the images for each t. The system produces four files containing the per-channel voxel information for $\{$red, green, blue, alpha$\}$, where our subsequent pipeline only makes use of the data from the alpha channel.

Each of the alpha channel volume files from the sequence is converted into a dense polygonal surface mesh using a version of Bloomenthal's Marching Cubes algorithm [10], previously modified by Quynh Dinh [29]. We subsequently modified the code that prefiltered the voxels to perform Gaussian filtering separately per dimension. We chose a prefilter kernel size of five to coincide with our voxel size, which, in turn, was chosen to model a subject's most spindly limbs with 10 voxels across. The result of this stage of processing is a sequence of dense, uniformly subdivided 3D triangle meshes representing the subject's changing shape over time.

Implementation of Spine Estimation: The implementation of our Spine estimation algorithm is actually a collection of smaller programs. These start with a sequence of polygonal meshes and finally produce a Spine graph parameterized on time. The key insight to the practical implementation of our algorithm in Chapter 3 is the use of a second graph structure. This second graph models the vertices of a polygonal mesh as nodes connected by edges weighted by the inter-vertex distances. Using the Library of Efficient Data Types and Algorithms (LEDA) [54] allowed us to efficiently traverse the graph structure, calculating geodesic distance using a fast implementation of Dijkstra's algorithm. The templated graph and node manipulation algorithms available in LEDA simplified the programming of level set computations, as well as file input and output for graph hierarchies. Our implementation

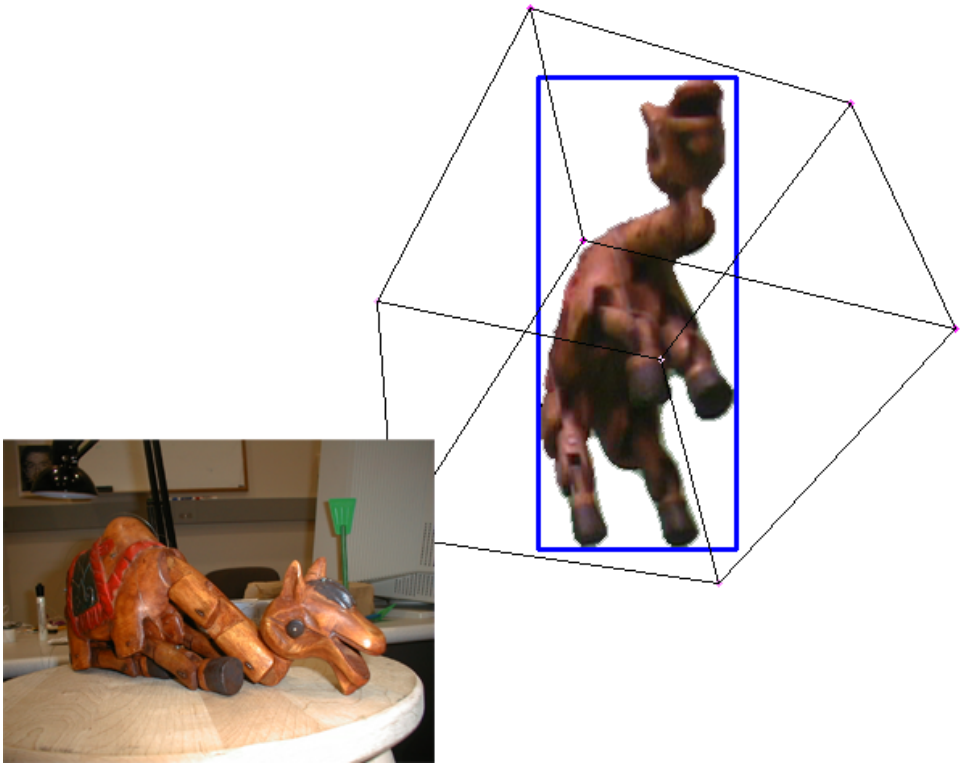


Figure 11: Camel marionette used for experimentation, after replacing strings with fishing line (inset). This segmented example frame from the video footage shows the blue 2D bounding box, and the subsequently estimated bounding volume used by the GVC algorithm.

of an OpenGL viewer for LEDA graphs has significantly simplified the development process, producing a stable and memory efficient, if not yet speed optimized, implementation of Spine estimation.

4.3 Results

Except for the tarantula and the synthetic example of the starfish, the subjects often required human proximity and were too heavy for the transparent flooring, so we were only able to leverage a subset of the cameras present.

Baby: The baby data is the result of filming an 11-month old infant using nine cameras. The sequence is 45 frames long (sampled at 30 frames per second) because that was the

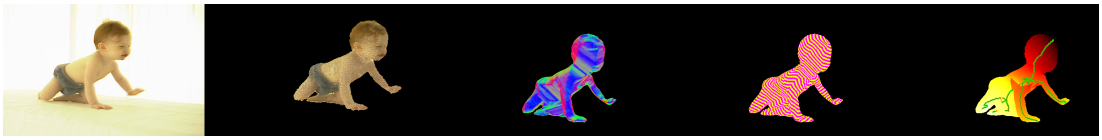


Figure 12: BABY DATASET: From left to right, one of the views, voxels, polygonal model, level sets, and Spine with distance function.

Figure 13: DOG DATASET: From left to right, subject, polygonal model, distance function, level sets, and resulting Spine.

Figure 14: CAMEL PUPPET DATASET: From left to right, one view, wireframe, distance function, level sets, and resulting Spine.

period she needed to crawl down the entire length of the stage. Her progress forward is mostly due to her arms and right leg, while she tends to drag her left leg, causing frequent merging of her voxel-model from the waist down. The Spine generation models her head and arms very consistently, but the correspondence tracker cannot resolve her legs and mis-assigns one leg or the other for the majority of frames (Figure 12).

Dog: The dog was the most challenging of the test-subjects simply because there were only seven cameras that could operate without also filming the dog's handlers. The 98 volume reconstructions are all close to their average of 1.04M voxels. Examination of the polygonal-mesh sequence reveals that much of this bulk comes from the ghost-voxels under his stomach that were carved successfully in the other test subjects when more cameras

were running (Figure 13).

Camel Puppet: The camel marionette, pictured in Figures 11 and 14, is 26 cm long and stretches to a height of 42 cm. While the subject certainly did not change in volume throughout shooting, its representation varied throughout the sequence between 600k and 800k voxels, largely due to self-occlusions. The polygonal representations averaged 200k polygons. The sequence is 495 frames (15 seconds) long, and was filmed using 12 color cameras. The camel's motion changes in the sequence from leg-jostling at the start to vigorous kicking and raising of the neck by the end. The system was only hindered by the repeated "merging" of legs as they tucked underneath or appeared close enough to each other to be joined in the voxel stage. As the most common such merging happened at the camel's back hoofs, the Spine generation succeeded in making limb tips at least close to the real limb tips. This estimate was sufficient for the exploratory Spine generation to feed the correspondence tracker, that in turn determined that there were five limbs. The number of limb tips found in each surface individually is plotted in Figure 15. While many frames contained six limb tips, the majority (and the largest consistent subsequences) had five and ignored the camel hump (see Figure 16). Applying the algorithm in Section 3.3 produces a sequence of skeletons whose limb correspondence is known. Applying the part of the algorithm described in Section 3.4 generates the parameterized Spine, allowing for the poses from the sequence to be played back through translation and rotation of the same geometry (see Figure 17). A resulting creature skeleton is pictured in Figure 7. As illustrated in Figure 7, the correspondence tracking balances out the greedy limb inclusion of the exploratory Spines.

Tarantula: The tarantula pictured in Figures 18 and 19 is a Chilean Rose-hair (*Grammastola Rosea*) and measures approximately 12*cm* in diameter. We filmed it on the transparent table surface, allowing all 20 cameras a clear view for six minutes. It walked deliberately, but in fits and starts, stopping occasionally for several seconds. We selected a 19 second

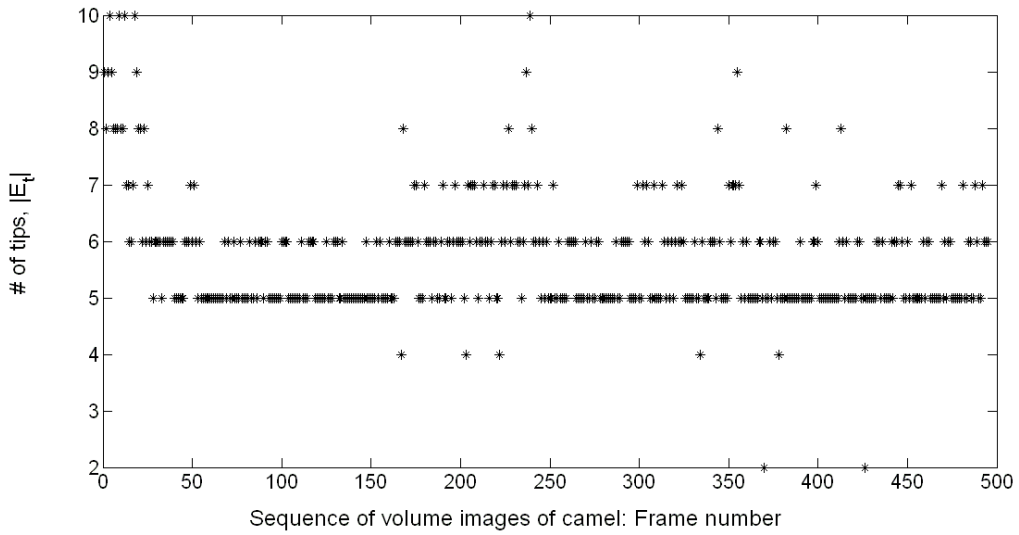


Figure 15: Graph of camel's number of end effector limb tips found over time when processing surfaces individually. $|E_t|$ varies between four and ten.

(570 frame) segment for processing, where there was almost continuous walking motion until the tarantula reached the edge of the table. Most of the surfaces were composed of approximately 400,000 triangles, produced by carving using the nine cameras that were both consistently calibrated and required no manual segmentation. The only cause for segmentation difficulties was the tarantula's reflection in the floor. Calibration and carving in general were difficult because the cameras were zoomed to the limit on a very small overlapping volume, and still only had the tarantula occupying 1.5% (100*50 pixels) of the total image. Cameras whose extrinsic calibration errors exceeded 0.2 pixels were excluded, and after subdivision of the volume, each voxel represented $\frac{1}{32}cm$, as compared to $\frac{1}{4}cm$ for our other live subjects. As a consequence, the resulting reconstructions appear "lumpy" (see Figure 20) because 2D segmentation boundaries carve large blocks of voxels. Although this carving allows an occasional limb to be carved off or abbreviated, the skeleton estimation proceeds to find most limbs in many of the surfaces of the sequence (see Figure 23). Figure 24 shows that most frames report finding 10 limbs – namely the eight legs and the two pedipalps (shorter sensing appendages on either side of the fangs).

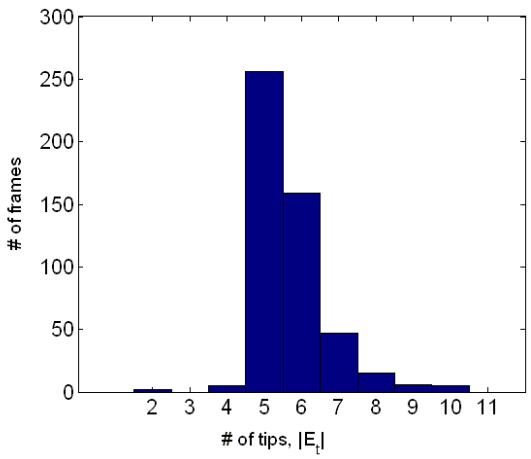


Figure 16: Histogram showing how many surfaces in the camel sequence were found to have each number of limb tips. The camel sequence revealed five limb tips with the greatest consistency, as is appropriate for this creature.

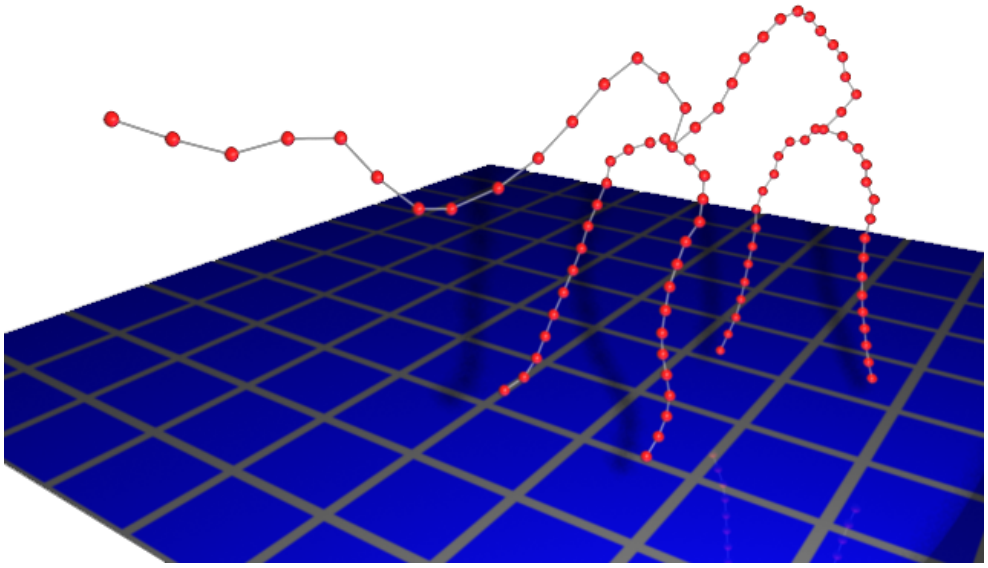


Figure 17: Rendered version of the camel Spine, extracted from the animated sequence. Pictured nodes appear consistently in each frame with known correspondence and orientation. Each limb was parameterized on length and discretized into 15 equal samples, though any subdivision can be used. Frames where limbs were tucked are interpolated for animation purposes.

Figure 18: Photograph of tarantula subject taken through transparent table top. Beyond eight legs, tarantulas have a pedipalp on each side of the fangs, and an abdomen section.

This estimate is less than ideal because arachnids actually have an 11th appendage, the abdomen, that was not located as consistently in our sequence. The temporal integration of the sequence of skeletons was not possible for the tarantula because of the combinatorial nature of correspondence tracking. Efficient multi-hypothesis testing for correspondence is a separate and interesting problem. A possible system-building extension to the algorithm for many-legged creatures can be imagined if the restriction on prior domain knowledge is lifted. One could use the heuristic that not all legs are in motion simultaneously, so many combinations of correspondence can be eliminated straight away just by finding the limbs that were motionless in a subsequence.

Adult Human: Visual Hull data of an adult human was kindly provided to us by Trevor Darrell's group at the AI Lab. The 62 Visual Hull polygonal surfaces were subdivided

Figure 19: Tiled subimages of tarantula subject's footage filmed with 20 cameras. Variations in colorsare result of user error in adjusting settings, though two images pictured in bottom row come from JVC cameras instead of Canon.

and resampled to eliminate the endemic long sliver triangles of that method. The subject was filmed with four cameras while punching air first with one arm, then the other (while tucking the first), and repeating. Because of the small number of cameras, the subject appears somewhat blocky (see right side of Figure 25), yet still reveals the pose of the legs, head, and alternating arms. Predictably, the tucking in of one arm or the other at all times leads the Spine estimation to generate only one arm – assigning correspondence of that limb to either the right or left arm, depending on which arm was extended at the time (see Figure 27). The resulting parameterized Spine is rendered on the left side of Figure 25. While the camel data was processed by the algorithm in Section 3.2 set to quantify geodesic distance into 100 level sets, all other surfaces including those of the adult human were segmented into 30 level sets. This lone argument to our algorithm, set using a coarse heuristic of mesh density per geometric distance, was altered to 40 level sets to generate graph in Figure 26B. Comparing it to graph 26A, we see that there were more frames where the fifth limb tip was found, owing to the fact that the hand on the tucked arm

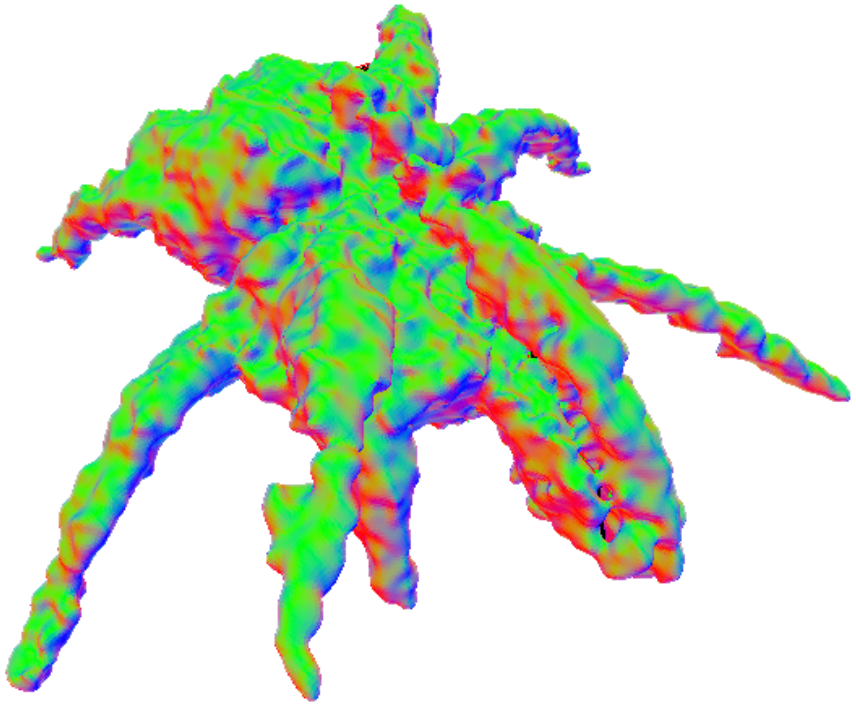


Figure 20: Reconstructed surface mesh of tarantula using array of frames #29. {Red, green, blue} coloring represents $\{ x, y, z \}$ components of surface normals. Lumps on surface are result of fine subdivision of voxel volume despite comparatively large pixels in video footage (at maximal zoom).

protruded enough to appear briefly as an end effector when the level set was narrower in geodesic distance space. This more dense subdivision of the surface still yields only four consistent limbs. Further increased numbers of level set subdivisions degenerate quickly because the sets' widths become the same as or narrower than the lengths of the polygonal edges – preventing a level set from completing a circuit around the surface.

Overall Performance: The average processing times for skeleton generation using our unoptimized implementation of the algorithms were consistently under two minutes per camel mesh on a Pentium 4 PC with one or more GB of memory. This and other polygonal data sets indicate that processing takes approximately one minute per 100,000 polygons. This number should be doubled to account for both the exploratory-skeleton generation and

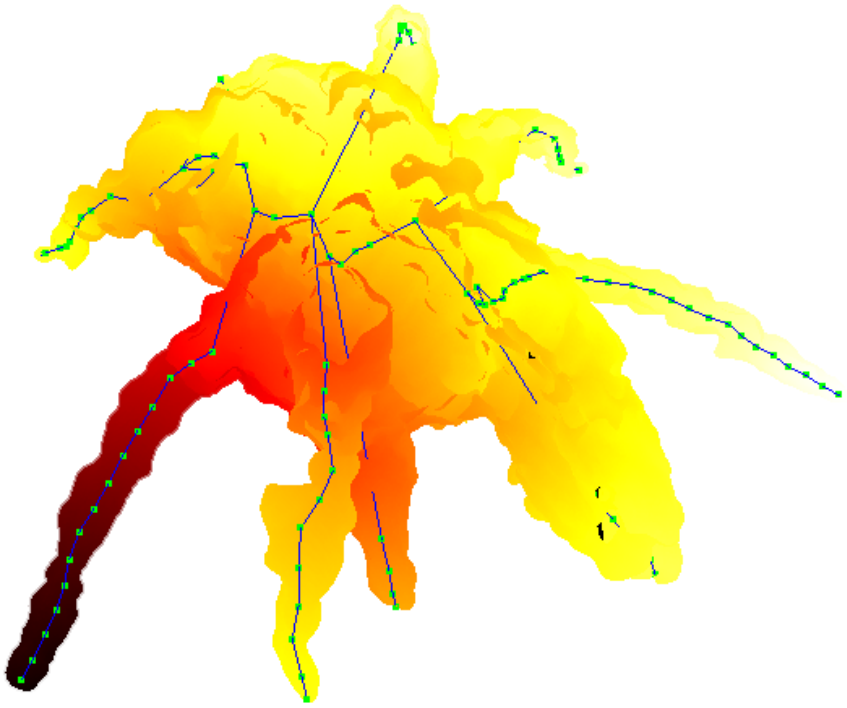


Figure 21: Geodesic distance measured from first automatically detected tarantula limb tip, colored from black to white with increasing distance. Subdivision of this distance field into connected components of level sets according to our algorithm produces the pictured skeleton. Other limb tips subsequently yield other skeletons that are merged with the pictured one.

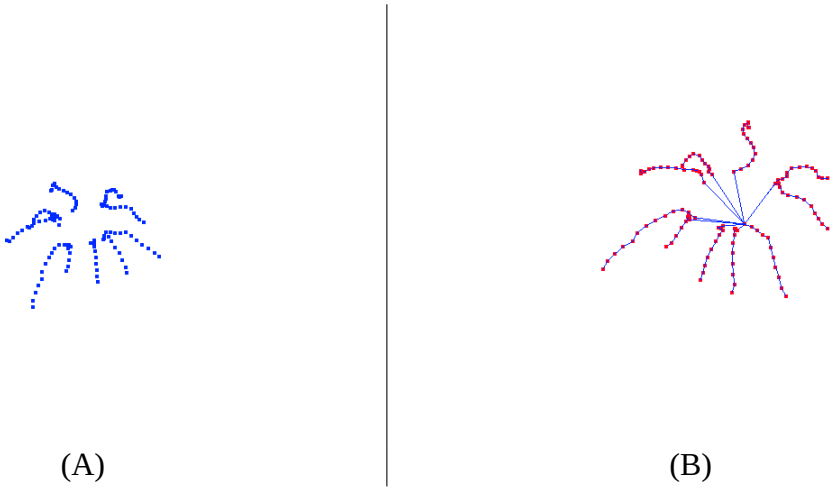


Figure 22: Different views of the merged tarantula skeleton: (A)Without edges connecting skeleton nodes, (B) with edges that converge on a point.

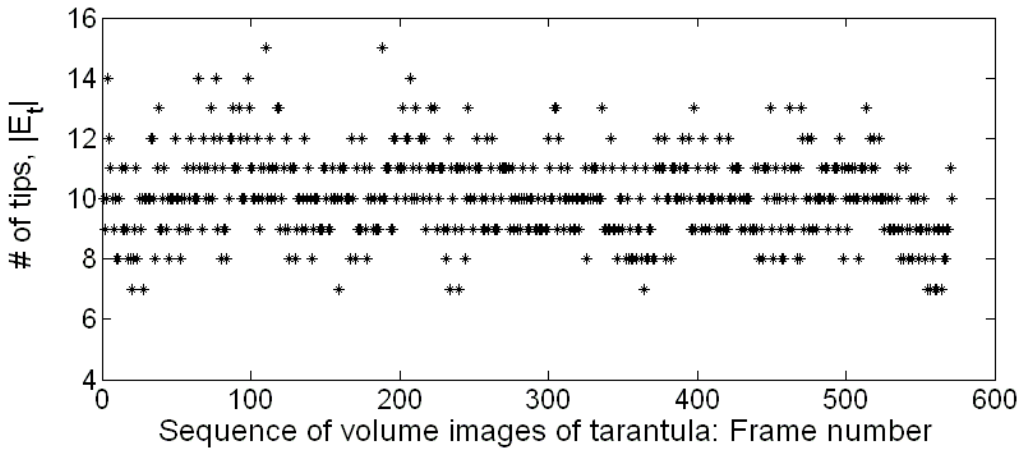


Figure 23: Graph of tarantula's number of end effector limb tips found over time when processing surfaces individually. $|E_t|$ varies between seven and fifteen.

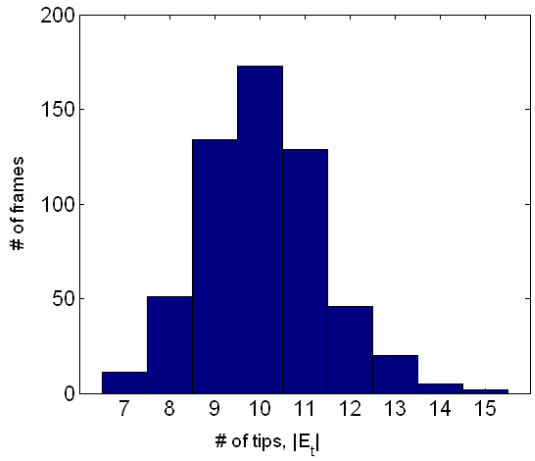


Figure 24: Histogram showing how many surfaces in the tarantula sequence were found to have each number of limb tips. The tarantula sequence revealed ten limb tips with the greatest consistency, which is almost appropriate; tarantulas have eight legs, two pedipalps, and an abdomen.

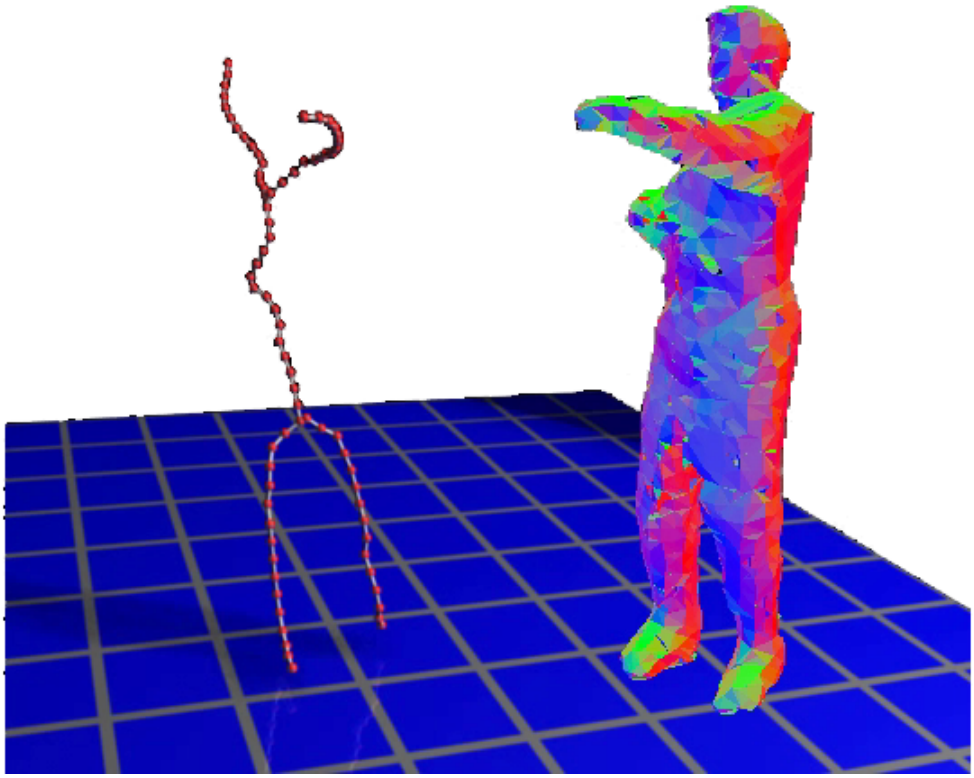
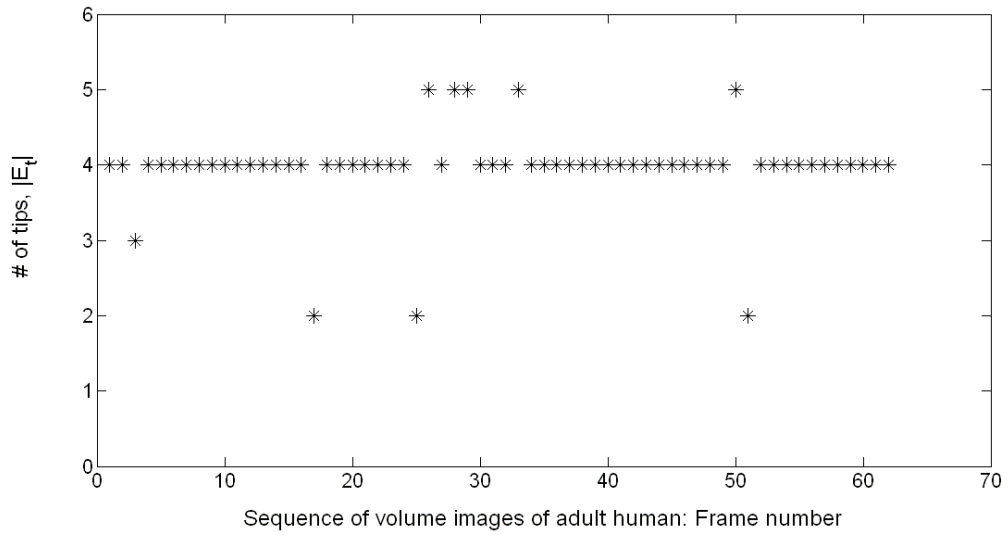
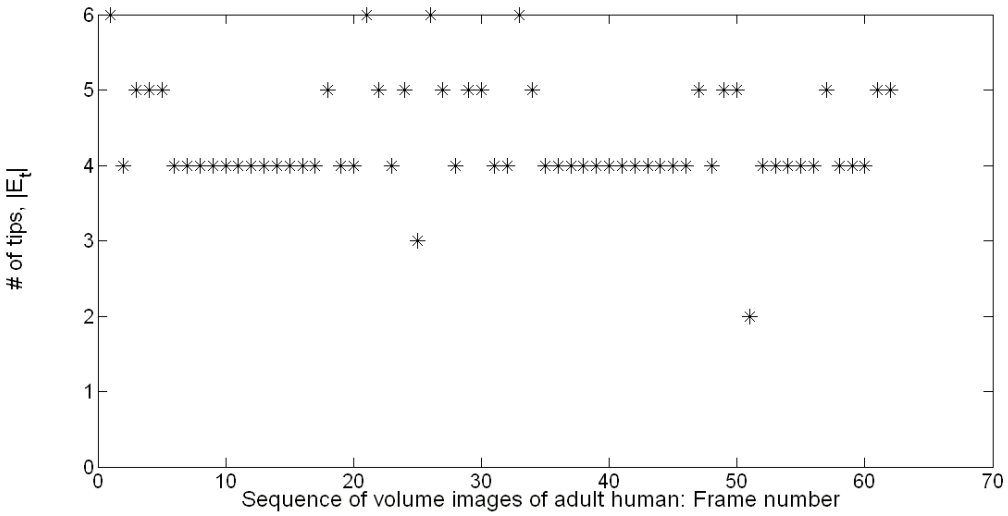


Figure 25: Rendering of Spine estimated for adult human subject (left), and corresponding surface mesh colored according to normals.

that performed after finding correspondence. The correspondence-tracking portion of the algorithm (Section 3.3) took ten minutes on the 495 frame camel sequence, and less than three minutes on the remaining processed sequences. Preprocessing to extract polygonal meshes from voxels averaged three minutes per frame on the tarantula, the largest volumes, that were embedded in $(448 * 224 * 448)$ voxels. The GVC carving of voxels from as many as 12 cameras took no more than 25 min., with an average of 14 min. per frame, and significantly less when fewer cameras were available.



A



B

Figure 26: (A) Graph of adult human's number of end effector limb tips found over time when processing surfaces individually. $|E_t|$ varies between two and five. (B) While most data was processed by quantizing the geodesic distance field into 30 levels, the pictured graph shows the change in $|E_t|$ when using 40 levels instead. The number of levels used cannot be arbitrarily high unless triangles in the surface mesh are subdivided, because level sets must form complete circuits around the body.

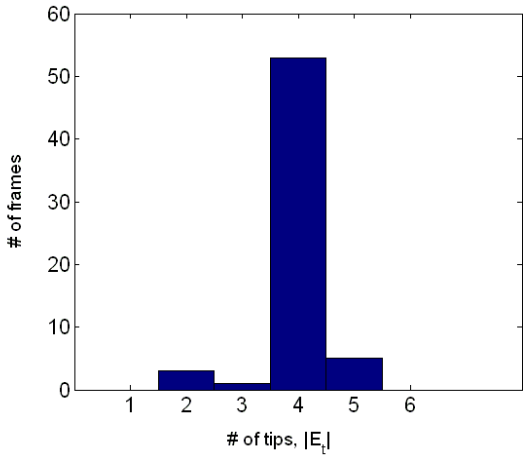


Figure 27: Histogram showing how many surfaces in the adult human sequence were found to have each number of limb tips. The adult human sequence revealed four limb tips with the greatest consistency instead of five, because one of the two arms was alternatingly tucked against the body.

4.4 Limitations

As discussed in Chapter 3, successful Spine estimation was possible subject to the following constraints:

Video Segmentation The subject had to contrast with its background or be otherwise segmentable to allow GVC to correctly carve its shape.

Camera Coverage Cameras were pointed at the subject in a manner that revealed gaps between limbs. In general, increased numbers of cameras produced superior volume reconstructions. Otherwise, limbs "disappeared" by merging their volumes with each other or the subject's torso.

Infrequent Loops For the geodesic distance metric to apply, meshes must have genus zero. Surfaces where a subject's end effectors touch or otherwise form loops can not be sliced into reliable level sets.

Demonstrate Articulations As seen in the adult human data, a subject that rarely reveals some of its appendages will be estimated as having fewer limbs than it really does.

Interestingly, without heuristics limiting the expected speed with which limbs move, the limbs need to be revealed simultaneously.

Limb Complexity As seen in the tarantula data, the algorithm's exhaustive search in aligning tree graphs representing the per-frame skeletons is limited for computational complexity reasons to subjects with less than ten limbs. To circumvent this limitation, heuristics about limb-velocity are needed to perform the limb matching correspondence search as an efficient form of multi-hypothesis testing.

CHAPTER V

COMPARISON OF APPLICABILITY

5.1 Possible Applications Overview

Aspects of several applications have the potential to be simplified and automated using Spine estimation. Existing domains that stand to benefit the most are ones where skeletons are employed but must be defined or posed manually. Even without the availability of a video or surface mesh sequence, the estimation of a branching skeleton (see Section 3.2) can be employed to save tedious user interaction.

One application that employs character skeletons deals with secondary simulation of elastic deformations. The problem in this case is how to respect the rigidity constraints imposed by the skeleton. Capell et al. [18] embed a single polygonal mesh inside of a coarse volumetric control lattice. The lattice defines the prominent bones and "meaty" parts of the character, enabling the authors to contribute an algorithm for performing locally linearized simulations of secondary motion. The same algorithm could be applied after a more automated character set-up. Given the one character mesh, Spine estimation could find the structure of the creature. The skeleton produced would have orientation information at each node in the limb hierarchy, that can be used to grow spokes that in turn define the control lattice. The manual version of that process took an experienced user hours for a moderately complex control lattice, but should be measurably faster when automated – likely on the order of a minute. Even if the node locations of the Spine need to be adjusted manually to afford specific animator rigging before the control lattice is synthesized, the process can still be expected to finish in much less than an hour.

Another interesting problem domain deals with animated deformations by example. A

common approach to parameterizing the relationship between example surfaces of character meshes makes use of limb pose. The work on deformation by example from range scan data by Allen et al. [1] shows a compelling application when both accurate pose and surface information is available for a human subject. While use of a range scanner results in detailed surface reconstructions of limbs, the estimation of their pose is heavily dependent on manual identification of colored dots that were painted on the subject's skin. 48 markers were painted on one arm and torso and subsequently identified in the different color images captured by the Cyberware scanner. 90 more were used for the torso. This process is a painstaking one, even when assisted by pose estimation algorithms – likely because marker alignment of multiple bodies is known to be susceptible to local minima. This difficulty is especially common when the limbs themselves are being optimized.

This system could possibly benefit from Spine estimation in two ways. First, after all the marker dots have been identified in one of the poses, their locations can be stored parameterized with respect to the skeleton estimated for that "frame." Subsequent poses can then use the relative locations of the dots to the Spine as a prior on correspondence and location. The second possible approach would require developing a mapping from the hand-made subdivision surface of the male subject to the Spine's limbs themselves. Such a mapping, based on limb lengths and Spine node orientations, could allow the marker dots to be skipped entirely, because their purpose is to assist with pose estimation in the absence of visual features. Again, Spine estimation would be using the volume of the subject as a collection of features containing similar pose information.

There is a broad range of applications that can benefit from the ability to find approximate poses without the use of explicit markers. Sand et al. [63] use a commercial motion capture system in conjunction with video cameras to acquire pose-dependent human body geometry. They explain however, that for this application, full skeleton estimation is unnecessary and complicates matters by introducing joint angle representation issues. Instead, they defined a table of limb segments and the collection of motion capture markers that

specifies the pose configuration of each limb. Section 5.2 below gives further details on how the limitations of indoor markered motion capture may be sufficiently overcome by Spine estimation to provide a viable alternative form of data capture for some situations.

5.2 The Motion Capture Problem

Addressing the matter of definition, ”motion capture” addresses the question of where features moved as time progressed. For 3D in general, motion capture is equivalent to the unconstrained 3D tracking problem. However, because most motion capture data is acquired for biomechanics or entertainment purposes, general purpose tracking is referred to as such. This nomenclature now leaves the term “motion capture” as usually referring to the specific finding of joint angles and end effector positions for articulated creature subjects.

For a single rigid object, it would be enough to define a single coordinate frame. For an articulated body however, it is necessary to define a hierarchy of coordinate systems, to uniquely express, for example, wrist rotation. Such a hierarchy is most conveniently represented as a collection of rigid limbs, attached to each other by revolute joints. For purposes of this discussion, we describe motion capture as seeking to measure the angles of all those revolute joints, *i.e.,* the pose.

5.3 Current Solution

The most accurate measurements of a creature’s pose are achieved by measuring the relative orientations of the bones with the use of bone screws. When three noncollinear screws protrude from the bone through the flesh of a limb, the 3D positions of their heads constrain that limb’s coordinate frame, (t, R). The noise inherent in any such observation means that the rigid body pose of a thusly markered limb is overconstrained (see Figure 28). The articulation of the system further constrains many limbs’ poses, allowing the use of only two or one such markers when their locations are modeled relative to the rest of the limb hierarchy (see Figure 29). Application specific optimizations are used to solve the

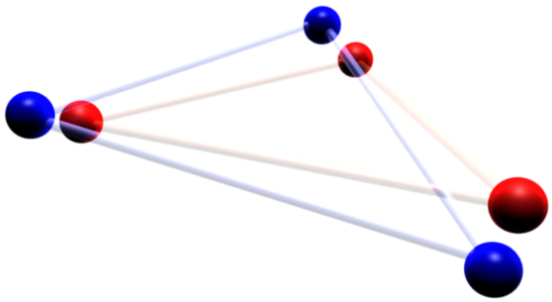


Figure 28: Example illustrating difficulty in aligning coordinate frames defined by three or more markers each. Co-locating one red and blue corner of the two frames can easily leave the other two corners grossly misaligned, so best fit techniques must be used instead (*e.g.*, least squared error on distance).

overconstrained poses.

For practical reasons, bone screws are replaced by active or passive surface markers in most commercial motion capture systems. While each marker of a magnetic system, such as the system built by Ascension [4], provides position and orientation information about a limb, the overall data interpretation is similar to that of tracked optical markers on a passive systems [72]. As markers affixed to the subject's surface can be occluded and slide with respect to the bones, extra markers are used, making the overconstrained poses less susceptible to such dropouts and marker slippage.

The difficulties with all passive marker systems are the need for (A) accurate triangulation in 3D, and (B) correspondence tracking over time. Various approaches exist, but one of the most effective ones [72] uses markers with retro-reflective tape. The markers are obvious and easy to track features that stand out compared to features one might find on clothing, skin, or fur. The tape is effective indoors when observed from a number (usually 12) of synchronized inward pointing cameras that have all but near infrared (IR) light filtered out. The near-IR light is beamed from batteries of LEDs surrounding each lens. The result for each calibrated camera is a binary image where markers appear as white dots, much like the moving light displays of Johansson [39]. The proprietary Vicon data capture software processes these calibrated images to produce 3D point clouds that retain

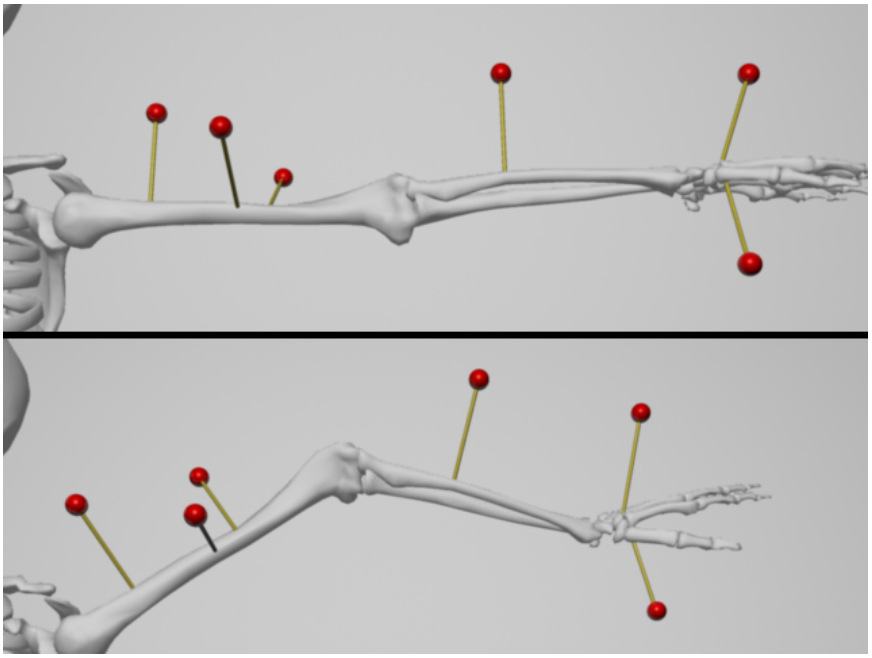


Figure 29: Illustration of attaching bone screw markers to a skeleton. Depending on prior knowledge about limb lengths and joint locations with respect to markers, not all limbs need three screws to be tracked effectively.

as much point correspondence over time as possible. Meaningful labels, correspondence errors (crossovers), and missing data (dropouts) are remedied manually as a data cleaning step.

As part of the post processing, the marker labels are used with the hierarchical model of the motion captured subject to assign the markers that drive each joint angle. Figure 30A illustrates an example armature and the *surface markers* that are explicitly assigned to drive the pose of each limb. Again, note that a limb's pose may be overconstrained by markers and articulation, so the final joint angles depend on the type of optimization being performed. As examples, the perfect fit of revolute joints may only appear as a weighted factor, in order to better minimize the mean squared error of distance between the model and the tracked markers, that themselves could be weighted to more strictly adhere to observed end effector positions.

5.4 Incorporating Spines

Spines deal with similar articulated creatures as existing motion capture systems. However, instead of relying on trackable surface features in the form of markers, Spines track the middle of limbs. Without the IR light limitations, Spines can also be tracked in creatures that cannot be markered and in some controlled outdoor settings (see Section 4.2). With these benefits of Spines, the expected cost of the markerless motion capture is increased uncertainty about the 3D location of Spine nodes – the discretized dense chain of points tracked along each limb. In pursuing a solution to the articulated motion capture problem, Spine nodes can be interpreted as internal markers, where each has a 3D position and orientation over time. Electromagnetic markers provide the same type of data, t and R, for each marker, but are active requiring a power source and are susceptible to slippage and interference from metal.

Driving an articulated skeleton is overconstrained when using Spine nodes just as it is when using optical markers. In the same fashion as with optical markers, a user or predefined heuristic could determine which Spine nodes drive each of the joints (see Figure 30B). While the locations of nodes suffice to determine a limb's translation, its rotation matrix must take advantage of other information, in case the nodes line up collinearly. The N_L nodes assigned to drive a given limb have individual orientations with respect to the body's home position where rotation was just the identity matrix. By fitting a gaussian to the N_L relative rotations, we can obtain a fairly robust joint angle for limb L in each frame.

5.5 Evaluation Plan

To evaluate the effectiveness of Spines in addressing the motion capture problem (Section 5.2), it is useful to compare them to the current state of the art. Optical motion capture is heavily engineered to deal with the particular settings and subjects where markers are convenient. Spines are meant to handle situations where markers are inconvenient, but are a more general purpose solution. Consequently, an evaluation of Spines as a method of

capturing motion could use optically markered motion capture (such as Vicon [72]) and an appropriately "simple" subject as ground truth. Possible example subjects include a dog, cat, large puppet, or with a large enough capture volume, a human.

After adjusting lighting in a motion capture studio with both synchronized motion capture (filtered) and color video cameras (unfiltered), calibration similar to that of Sand et al. [63] will bring the two coordinate frames into alignment. Optical markers should be attached to the subject in a manner consistent with the optical motion capture system, though flat markers should be used where possible to avoid altering the silhouettes. With non-directional lighting, the video cameras will see the markers as plain gray tape. If the subject is cooperative, they should stand in a T-pose that will serve as a reference pose for both systems. The subject can perform a variety of movements, preferably exploring their range of motion.

The resulting optical 3D marker tracks should be manually assigned to the limbs they will drive. The same should be done with the Spine nodes. A known pose estimation algorithm should be used to calculate joint angles (relative to the reference frame) for a CG model of the subject, where the model can be hand made from either manual limb length measurements, or from Spine estimates of limb lengths [1]. The simplest optimization would minimize the mean squared error of the CG model's virtual markers, both optical and Spine nodes, to the respective real data streams, as described in Sections 5.3 and 5.4. Acknowledging that optical motion capture is itself not free of measurement noise, we propose using the joint angles it calculates as ground truth. The joint angles produced from Spine data can be compared to those in terms of angular distance. Another metric should compare the 3D distance between end effectors of the CG model as it plays back the poses as estimated by both systems.

[1] A secondary form of evaluation could apply the work of O'Brien et al. [58] to both motion capture markers and Spine nodes to calculate limb lengths and joint locations. The Spine data, having volume information, would have an advantage in finding joint locations inside the body. Poorly exercised joints (elbows, knees) tracked with regular motion capture often reveal only an axis on which the joint lies.

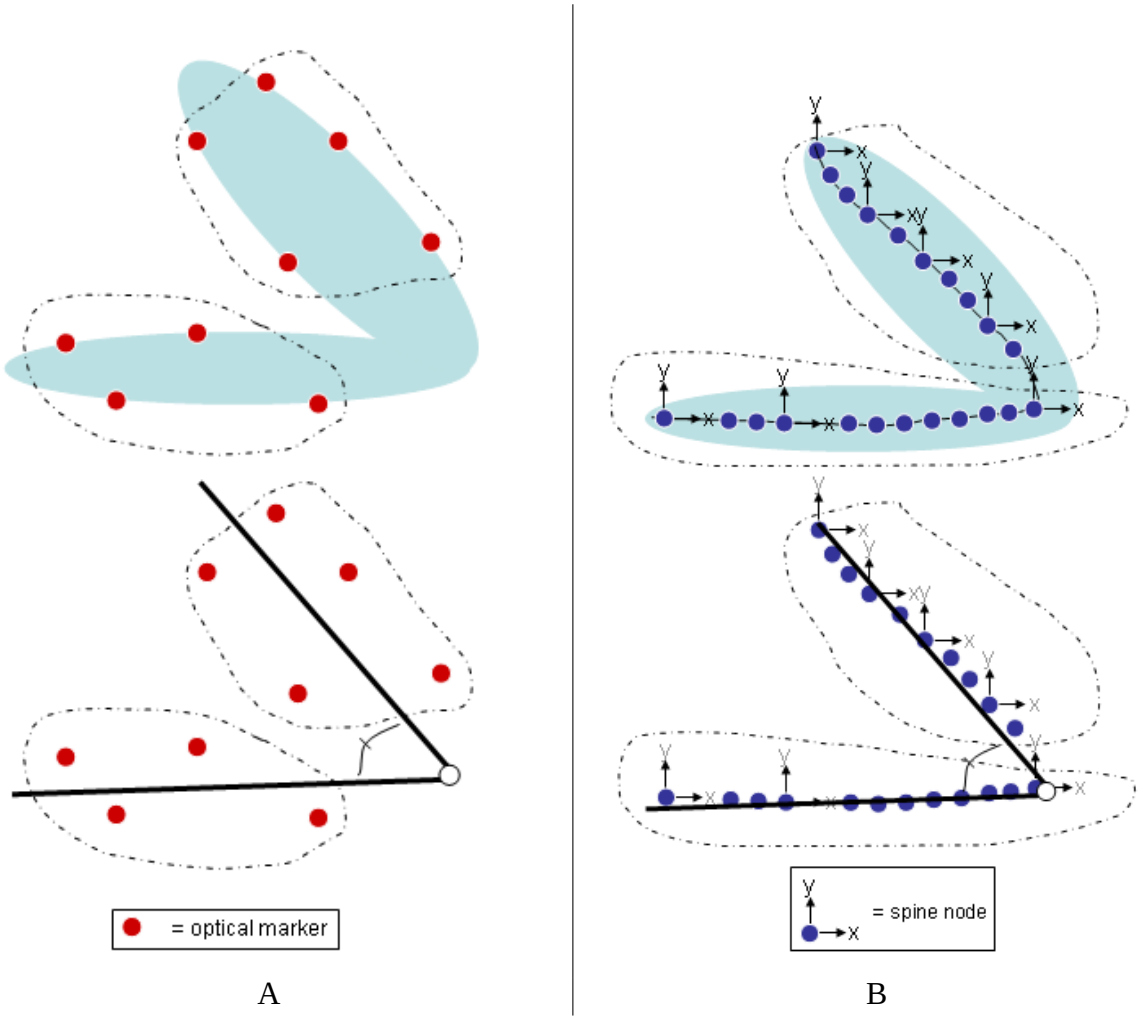


Figure 30: (A) Illustration of explicit optical motion capture markers attached to subject's surface (top). In postprocessing, a user selects clouds of markers that subsequently drive the pose of each limb in a revolute joint armature (bottom). (x,y,z) location of each optical marker is tracked and available for pose estimation (B) Illustration of implicit Spine node markers in subject's interior. In the same manner as for motion capture, a user can select clusters of nodes to drive each of the two pictured limbs on an armature. Spine node markers have (x,y,z) location and absolute orientation.

CHAPTER VI

CONCLUSION & FUTURE WORK

We have proposed *Spines* as a novel 3D spatio-temporal representation for sequences of volume images. This shape and motion descriptor introduces a method for imposing temporal correspondence on limb topologies when dealing with articulated subjects. We also present an algorithm for efficiently extracting branching Spines from surface data. Finally, we have presented example data where the temporally integrated canonical graph improves the quality of individual skeletons.

6.1 Contributions

This work is concerned with bridging the gap between surface geometry and functional structure. Partly out of necessity, this problem has led us to a combined data representation of surface *with* structure *as a function* of time. Previous efforts have generally addressed two but not all three of these metrics.

While this approach is rather data-intensive, it illustrates an automatic and non-invasive means of building flexible structural models of living creatures. Our three primary contributions follow.

> **I.** A functional structure emerges completely automatically for creatures that reveal their articulations – given no prior information.

Model-based tracking is now an established area of research in computer vision, but the underlying model is often assumed as given by a user. There is a range of possible internal mechanisms that could support the motions performed by a creature. Eventually, the ideal might be to reconstruct the creature's real osteology only from such observations. As a start,

we have at least shown that a Spine, or flexible branching armature, can be determined repeatably and automatically, given only visual observation data. One advantage such a data-driven algorithm might offer is that subsequent model-based tracking algorithms could be compared to each other more fairly by starting with the same limb-hierarchy and limb-length prior, independent of a user's skill or intuition.

II. The Spine representation allows for probabilistic integration of spatial and temporal data.

For the specific class of surfaces representing articulated creatures, the logical progression from the 2D skeleton to 3D has been extended to incorporate the dimension of time. Our approach accomplishes the first iteration of estimating a single limb hierarchy for a data sequence and projecting that hierarchy back on the individual surfaces, maximizing our spatio-temporal objective function. Our objective function seeks to represent moving articulated creatures by tracing the middle of their limbs (center of the "meat") that appear consistently as appendages. Other objective functions should be examined for general use, and for subjects with other specific characteristics [75]. This area of research is fairly new because the availability of sequential volume or surface data is a recent development, and poses many questions.

III. The search for correspondence of 3D features can be performed in lower dimensions by imposing the consistent articulation constraint.

Whatever the original dimensionality of input data, incorporating time requires that the question of correspondence be addressed. In a single video, this task amounts to optical flow. In our multi-camera videos of motion, tracking correspondence requires locating each pixel's corresponding location in the other videos, or determining that it has been occluded. Instead, tracking correspondence in volumes or on surfaces makes the task independent of the number of cameras. However, even when reduced to polygonal mesh surfaces, usually

on the order of 10*e*5 polygons, repeatedly locating the same 3D features throughout the sequence is computationally complex. We impose the constraint that the articulated creature has a consistent limb hierarchy. We have experimentally shown that we can therefore just locate the same limbs in each frame, which reduces to the matching of tree graphs.

6.2 Future Work

Where the current fully bottom-up work leaves off, extensions are planned that will allow a prior skeleton estimate to be forced on the data. This prior will especially apply to meshes where the limbs tuck in or become genus 1+. While the current results reflect that fairly noisy data, without priors, still reveals the real end effectors and underlying structure, further work is needed to track pose even in very poor data. The algorithm presented here works on regular and dense meshes of genus zero, but could be adapted to incorporate existing continuous geodesic distance algorithms.

True expectation maximization (EM) requires iteration until convergence. The current approach requires only the first iteration to converge on a limb hierarchy that is consistent with the data. However, the per-mesh location of the Spine could be a separate convergence criterion. By adding a procedure to project *G* probabilistically *back* into the surface mesh of each frame, EM could continue improving the placement of Spine nodes until their location only changes by some threshold.

The quality of the captured surface models is dependent on sufficient camera coverage, and is currently limited to creatures that truly reveal their degrees of freedom. However, one direction for further research is examination of predictive tracking mechanisms that could suggest when volumetrically-adjacent elements should be considered separate limbs. This direction could allow for more robust estimation of pose, or for tracking with fewer cameras. The data we have already obtained could be reprocessed, leaving out subsets of cameras to quantify the contribution of each viewing angle.

There is a need to establish a measure for how “bad” a degenerate frame is. Undesirable

degeneracies can occur due to persistent protrusions masquerading as limbs, looped limb poses, and poor camera coverage. Tracking through such degenerate poses is our primary area of further research.

6.2.1 Fusing Spine Nodes

The Spine structure generated by this approach significantly simplifies the parameter space of controlling a character mesh for tracking or animation, but still leaves too many unnecessary degrees of freedom. Interesting conclusions can be drawn from flexibility-analyses and minimum-energy optimizations of these Spines. For example, such analysis could better reveal which sections of a surface mesh require denser remeshing than the planar-sweep approach frequently used when axial data is available.

While our Spines can be used to synthesize mocap-like data, they will further be useful in automatically building a new functional *skeleton* with revolute joints. The Spine segments are a mid-level discretized representation of the surface geometry. A functional skeleton is higher-level still, with just the DOFs that were exhibited by the moving subject, only indirectly dependent on surface geometry.

This subsection explores the matter of skeleton estimation from Spines – the flexion goal. The motion of the observed Spine could be played back through a variety of skeletons, each with different numbers of DOFs, modeled here by revolute joints (3 DOFs each). The upper bound on skeleton complexity is to represent each inter-node link in the Spine as a small bone. This dense sampling could allow for nearly "perfect" playback of the captured data, inclusive of all the DOFs except intra-node translation that would require bone-stretching. The lower bound on skeleton complexity is the trivial case of representing all the nodes in a Spine as a rigid point-cloud fused around one bone.

Given specific input Spine data, we can expect to automatically generate a functional skeleton that lands between these two extremes of skeleton complexity while maximizing

the retained range of motion (see Sec. 6.2.2 below). The Spines have all the DOFs necessary to reach the different poses observed in the 3D input sequence, but also contain many DOFs that were never exercised. Such Spines are hard to integrate into most vision-based tracking algorithms, and are quite challenging for tasks where new key-frame poses would be synthesized (like the predicted/interpolated poses necessary for segmentation). For these reasons, skeletons that retain only the range of motion necessary for *specific applications* are desirable.

6.2.2 Limb Optimization Frameworks

The Spines were synthesized in a bottom-up data-driven fashion that introduced no changes to the acquired sequence of surfaces. Given a technique for remeshing the Spines and rendering them with texture information, we would expect zero-error reprojections when comparing to the original video footage. As we decrease a Spine's range of motion by approximating it with a limited DOF skeleton, we would expect progressively larger reprojection errors. Ideally, the fitting of skeletons to our motion data would be optimized relative to such an image-error metric, but that approach is currently prohibitive; We can find no existing sufficiently principled way of remeshing limb junctions, and factoring out the influence of limb-pose vs. action-specific dynamics on surface deformations is beyond the scope of this work.

We would instead choose the 3D distance metric from (a) the Spine-nodes on the time-varying desired Spine to (b) trajectories of nodes that have been fixed relative to the skeleton being optimized. The possible optimization variables (subject to a set error threshold) are

- Number of joints, j
- Locations of joints relative to branches of the Spine-tree
- Limb orientations referred to as joint angles, R_j

The practical challenge of optimizing all of these variables jointly is significant because of the combinatorial increase in dimensionality as j grows. Once placed, a joint has only

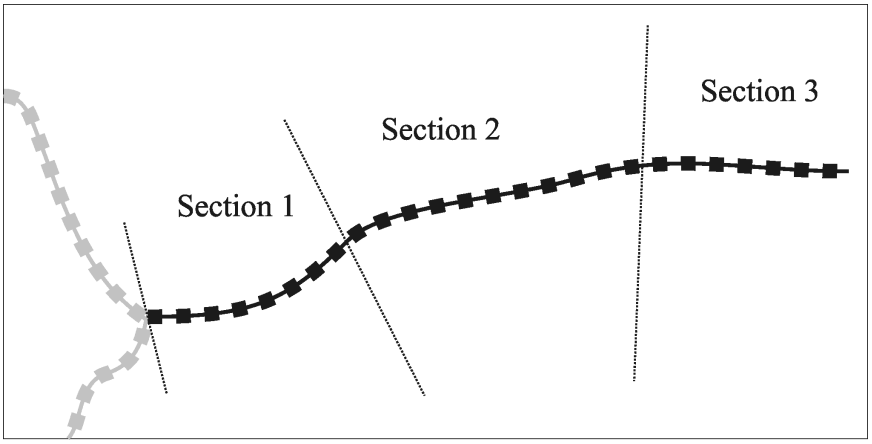


Figure 31: One hypothesized separation of this branch of the Spine tree into three sections, each being evaluated as rigidly rotating about its parent joint.

the three rotational degrees of freedom, but given N nodes in a Spine, there are N choose j combinations of joint locations if we limit joints to occur only at sites of Spine nodes. For conservatively selected data with 300-node spines and fixing j to only 10 joints, this search amounts to 10^{18} combinations of joint locations.

To make this optimization tractable, we propose testing out two pose-estimation strategies. Both make the simplifying assumption that the upper bound on the number of joints *per branch of Spine nodes* can be determined using some local flexibility metric. Both optimizations proceed by comparing hypothesized per-branch Spine node boundaries (Figure 31).

The two pose-estimation strategies are based on the previous work of O'Brien et al. [58] and Weik & Liedtke [74] respectively. For the first strategy, the nodes constituting each supposedly rigid section of a limb are treated as having a single translation and orientation per-frame of the sequence, thereby fitting into the O'Brien et al. [58] optimization designed for magnetic motion capture. The drawback of this technique is that while the joint locations are optimal given limb-trajectories, there is no co-location guarantee on a joint as computed for its parent vs. child limbs - a joint can fly apart, though this disagreement will be reflected in the residual error.

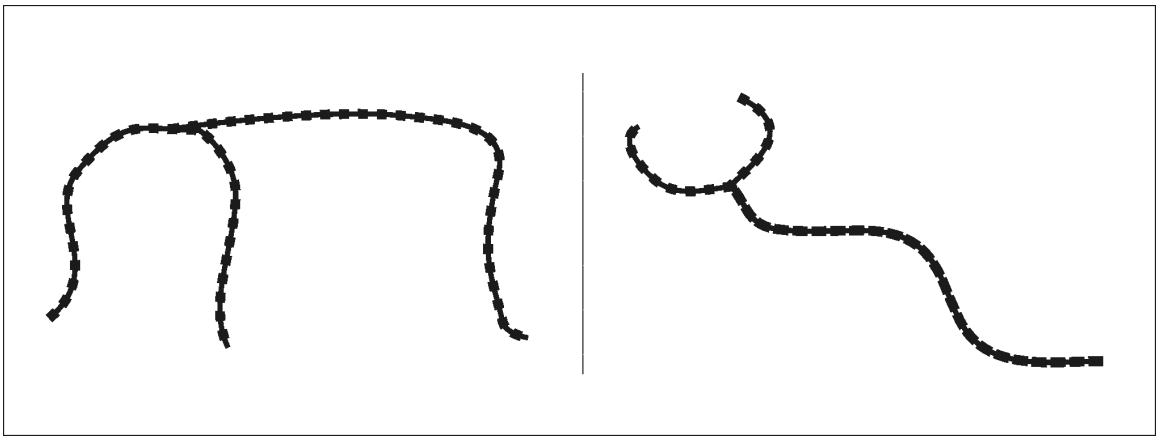


Figure 32: While both are Spine trees appearing to have three limbs, the creature on the left has three legs, while the one on the right is a snake with bull-horns. Performing a local Spine flexibility analysis would reveal a good starting point for the placement of three joints in either case, and reduces superfluous joint-location hypotheses.

The pose-estimation strategy based on Weik & Liedtke [74] also starts with hypotheses of Spine-branch partitioning. To hierarchically estimate the pose of each limb in a limb-tree, we apply their version of the Iterated Closest Points (ICP) algorithm [7]. By starting at the root of the hierarchy and working our way outward to find the pose of each subsequent limb, the joints are defined as co-located because the translational DOF of ICP is disabled. The drawback here is that error can accumulate as we get closer to estimating nodes near the end-effector.

These comparatively smaller optimizations sample different hypotheses of joint-placements within each branch of a Spine tree. Factored sampling of these hypotheses will essentially perform a hill-climbing through the space of possible joint-locations. Figure 32 illustrates the need for local flex-analysis that would provide a heuristic prior for this space. Note that the this prior provides an opportunity for user interaction, in that "preferred" joint locations can be suggested by a person examining the Spine tree. Further, the error of each joint-location hypothesis (from the associated pose-estimate sequence) can be augmented to include a smoothness metric that would discourage abrupt pose-changes in the sequence, if such heuristic data is deemed appropriate for some target application.

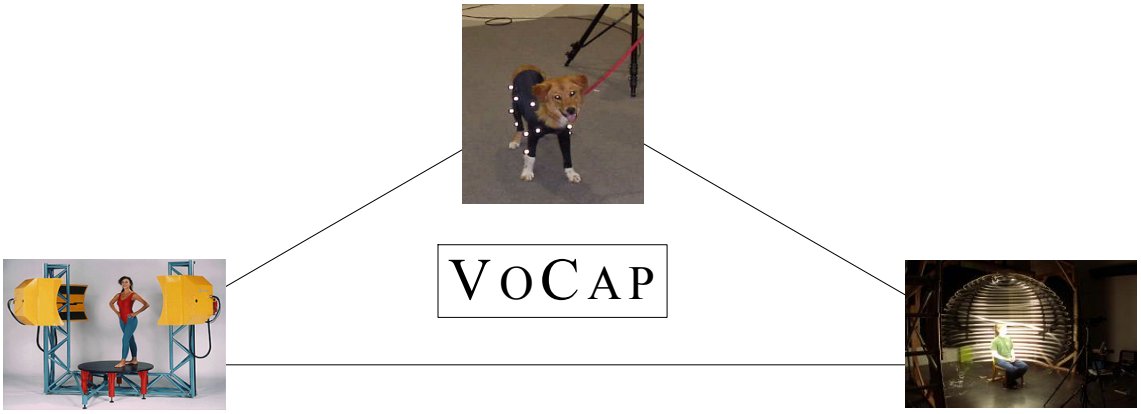


Figure 33: Volume Capture merges technologies of data acquisition for skeletal poses (mocap), surface models (Cyberware), and lighting/texture ([26]).

6.2.3 Use of this Representation

As with all good data representations, the purpose of ours is not solely to establish a standard for information storage [1]. When this research area proceeds, there should be a marked improvement in the ability to perform data-driven analysis of human and animal movement. While current motion capture systems suffice for many creatures [37], more detailed biomechanical study will be possible once marker-free data of volume sequences (VOCAP) can be analyzed (*e.g.*, insects, octopi) [31]. Factoring out the rigid-body transformations of limbs will also make possible the study of pose-dependent and dynamics-dependent surface and muscle deformation.

Furthermore, VOCAP data itself could become a canonical type used to gauge algorithms and implementations of systems for capturing articulated movement of creatures. The systems of Chu et al., Vedula et al., Cheung et al., and Borovikov & Davis [20, 70, 19, 12] are capable of capturing multi-view video data and processing the sequences of volumes. However, because they address specific application goals and use vastly different

[1] We hope to eventually mirror Marr's criteria for shape recognition representations: accessibility, scope and uniqueness, and stability and sensitivity [52]

prior knowledge, it is currently difficult to compare them fairly. More researchers are increasingly gathering and examining this type of data (researchers on Rossignac et al. [62] have but a few test cases) by attempting to build capture systems to supplement mocap systems, and VoCap can be expected to be of benefit in many such domains. We have started sharing our preliminary test data, and already there is significant demand for that from several research groups.

6.3 Final Discussion

We are inspired by Marr's theories on representations of shape for recognition [52]. His "3-D Model Representation" of nested primitives relies on the presence of natural axes in subjects' limbs. Just as his modular organization allows scope to be traded against detail in three dimensions, we hope similar theories can now emerge with the fourth dimension of time. To incorporate time, we have proposed Spines as an algorithmic approach to dealing with this new kind of data. As the technology used to acquire the data changes, possibly skipping currently necessary reconstruction stages, we hope the joint representation of shape and motion will mature as well.

REFERENCES

[1] ALLEN, B., CURLESS, B., and POPOVIĆ, Z., "Articulated body deformation from range scan data," *ACM Transactions on Graphics*, vol. 21, pp. 612–619, July 2002.

[2] ALLEN, P. K., ROSS, K. A., MESKELL, L., FEINER, S. K., and MURRAY, S., "Computational tools for modeling, visualizing and analyzing historic and archaeological sites." NSF-ITR Grant #0121239, 2001.

[3] ARUN, K. S., HUANG, T. S., and BLOSTEIN, S. D., "Least squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, pp. 698–700, March 1987.

[4] ASCENSION, *Ascension Technology Corporation*. http://www.ascension-tech.com, 2004.

[5] ATTALI, D. and MONTANVERT, A., "Computing and simplifying 2d and 3d continuous skeletons," *Computer Vision and Image Understanding*, vol. 67, pp. 261–273, September 1997.

[6] BERNARDINI, F. and RUSHMEIER, H., "The 3d model acquisition pipeline," *Computer Graphics Forum*, vol. 21, no. 2, pp. 149–172, 2002.

[7] BESL, P. and MCKAY, N., "A method for registration of 3d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 1992.

[8] BETELU, S., SAPIRO, G., TANNENBAUM, A., and GIBLIN, P. J., "Noise-resistant affine skeletons of planar curves," in *ECCV00*, pp. I: 742–754, 2000.

[9] BINFORD, T., "Generalized cylinder representation," in *Encyclopedia of A. I.*, pp. 321–323, John Wiley & Sons, 1987 (first presented in 1971).

[10] BLOOMENTHAL, J., "An implicit surface polygonizer," in *Graphics Gems IV*, pp. 324–349, Academic Press Inc., 1994.

[11] BLUM, H., "Biological shape and visual science (part I)," *Journal of Theoretical Biology*, vol. 38, pp. 205–287, 1973.

[12] BOROVIKOV, E. and DAVIS, L., "3d shape estimation based on density driven model fitting," in *Proceedings of The 1st International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, IEEE, 2002.

[13] BRADKSI, G. and PISAREVSKY, V., "Intel's computer vision library: Applications in calibration, stereo, segmentation, tracking, gesture, face, and object recognition," in *Proceedings of IEEE Computer Vision and Pattern Recognition Conference 2000*, vol. II, pp. II:796–797, 2000. Demonstration Paper.

[14] BRAUN, M., *Picturing Time: The Work of Etienne-Jules Marey (1830-1904)*. University of Chicago Press, reprint ed., 1994.

[15] BREGLER, C. and MALIK, J., "Tracking people with twists and exponential maps," in *CVPR98*, pp. 8–15, 1998.

[16] CAO, Y., *Axial Representations of 3D Shapes*. PhD thesis, Brown University, 2003.

[17] CAO, Y. and MUMFORD, D., "Geometric structure estimation of axially symmetric pots from small fragments," in *Proc. IASTED SPPRA*, June 2002.

[18] CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., and POPOVIĆ, Z., "Interactive skeleton-driven dynamic deformations," *ACM Transactions on Graphics*, vol. 21, pp. 586–593, July 2002.

[19] CHEUNG, G. K., BAKER, S., and KANADE, T., "Shape-from-silhouette of articulated object and its use for human body kinematics estimation and motion capture," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, vol. 1, pp. 77–84, 2003.

[20] CHU, C., JENKINS, O., and MATARIC, M., "Markerless kinematic model and motion capture from volume sequences," in *CVPR03*, pp. II: 475–482, 2003.

[21] CORMEN, T. H., LEISERSON, C. E., and RIVEST, R. L., *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

[22] CULBERTSON, W. B., MALZBENDER, T., and SLABAUGH, G., "Generalized voxel coloring," in *ICCV Vision Algorithms Workshop*, no. 1883 in LNCS, pp. 100–115, Springer-Verlag, Sept. 1999.

[23] CYBERWARE, *Cyberware Whole Body Color 3D Scanner*. http://www.cyberware.com, 2003.

[24] DARRELL, T., "Vision interface group." http://www.ai.mit.edu/people/trevor/vision-interface.html, 2004.

[25] DAVISON, A. J., DEUTSCHER, J., and REID, I. D., "Markerless motion capture of complex full-body movement for character animation," in *Proceedings of Eurographics Workshop on Computer Animation and Simulation*, Sep 2001.

[26] DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., SAROKIN, W., and SAGAR, M., "Acquiring the reflectance field of a human face," in *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pp. 145–156, July 2000.

[27] DEUTSCHER, J., BLAKE, A., and REID, I., "Articulated body motion capture by annealed particle filtering," in *CVPR00*, pp. II: 126–133, 2000.

[28] DEY, T. K. and ZHAO, W., "Approximate medial axis as a voronoi subcomplex," in *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications*, pp. 356–366, ACM Press, 2002.

[29] DINH, H. Q., "Homepage." http://www.cs.stevens-tech.edu/ quynh/, 2004.

[30] FERLEY, E., CANI, M.-P., and ATTALI, D., "Skeletal reconstruction of branching shapes," *Computer Graphics Forum*, vol. 16, pp. 283–293, December 1997. Published under the name Marie-Paule Cani-Gascuel.

[31] FULL, R. J., KUBOW, T., SCHMITT, J., HOLMES, P., and KODITSCHEK, D., "Quantifying dynamic stability and maneuverability in legged locomotion," *Integ. and Comp. Biol*, no. 42, pp. 149–157, 2002.

[32] GAVRILA, D. M., "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 1, pp. 82–98, 1999.

[33] GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., and COHEN, M. F., "The lumigraph," in *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pp. 43–54, Aug. 1996.

[34] HASTIE, T. and STUETZLE, W., "Principal curves," *Journal of the American Statistical Association*, vol. 84, pp. 502–516, 1989.

[35] HILAGA, M., SHINAGAWA, Y., KOHMURA, T., and KUNII, T. L., "Topology matching for fully automatic similarity estimation of 3d shapes," in *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pp. 203–212, August 2001.

[36] HUBBARD, P. M., "Approximating polyhedra with spheres for time-critical collision detection," *ACM Transactions on Graphics*, vol. 15, pp. 179–210, July 1996.

[37] HUTCHINSON, J. R., FAMINI, D., LAIR, R., and KRAM, R., "Biomechanics: Are fast-moving elephants really running?," *Nature*, vol. 422, pp. 493–494, 2003.

[38] IKEUCHI, K., "The great buddha project," in *IEEE ISMAR03*, IEEE, 2003.

[39] JOHANSSON, G., "Visual motion perception," *Scientific American*, vol. 232, no. 6, pp. 76–88, 1975.

[40] JU, S. X., BLACK, M. J., and YACOOB, Y., "Cardboard people: A parameterized model of articulated motion," in *International Conference on Automatic Face and Gesture Recognition*, (Killington, Vermont), pp. 38–44, 1996.

[41] KATZ, S. and TAL, A., "Hierarchical mesh decomposition using fuzzy clustering and cuts," *ACM Transactions on Graphics*, vol. 22, pp. 954–961, July 2003.

[42] KÉGL, B., KRZYŻAK, A., LINDER, T., and ZEGER, K., "Learning and design of principal curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 281–297, 2000.

[43] KOUDELKA, M. L., BELHUMEUR, P. N., MAGDA, S., and KRIEGMAN, D. J., "Image-based modeling and rendering of surfaces with arbitrary brdfs," in *2001 Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, pp. 568–575, Dec. 2001.

[44] KUTULAKOS, K. N. and SEITZ, S. M., "A theory of shape by space carving," *International Journal of Computer Vision*, vol. 38, pp. 199–218, July/August 2000.

[45] LEVOY, M. and HANRAHAN, P. M., "Light field rendering," in *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pp. 31–42, Aug. 1996.

[46] LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., and FULK, D., "The digital michelangelo project," in *Proceedings of SIGGRAPH 2000* (AKELEY, K., ed.), Computer Graphics Proceedings, Annual Conference Series, (New York), pp. 131–144, ACM, ACM Press / ACM SIGGRAPH, 2000.

[47] LEWIS, J. P., CORDNER, M., and FONG, N., "Pose space deformations: A unified approach to shape interpolation and skeleton-driven deformation," in *Siggraph 2000, Computer Graphics Proceedings* (AKELEY, K., ed.), ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[48] LEYMARIE, F. F. and KIMIA, B. B., "The shock scaffold for representing 3d shape," in *Visual Form 2001* (C. ARCELLI, L.P. CORDELLA, G. S. D. B., ed.), no. LNCS 2059 in Lecture Notes in Computer Science, pp. 216–229, Springer-Verlag, 2001.

[49] LI, X., TOON, T. W., and HUANG, Z., "Decomposing polygon meshes for interactive applications," in *Proceedings of the 2001 Symposium on Interactive 3D graphics*, pp. 35–42, ACM Press, 2001.

[50] MALZBENDER, T., GELB, D., and WOLTERS, H., "Polynomial texture maps," in *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pp. 519–528, August 2001.

[51] MARR, D. and NISHIHARA, H., "Representation and recognition of the spatial organization of three-dimensional shapes," in *Proc. of the Royal Society of London, series B*, vol. 200-1140, pp. 269–294, February 1978.

[52] MARR, D., *Vision. A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman & Company, 1982. MAR d 82:1 1.Ex.

[53] MATUSIK, W., BUEHLER, C., RASKAR, R., GORTLER, S. J., and MCMILLAN, L., "Image-based visual hulls," in *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pp. 369–374, July 2000.

[54] MEHLHORN, K. and NÄHER, S., *LEDA – A Platform for Combinatorical and Geometric Computing*. Cambridge University Press, 1999. P LED 99:1 1.Ex.

[55] MIKIC, I., TRIVERDI, M., HUNTER, E., and COSMAN, P., "Articulated body posture estimation from multi-camera voxel data," in *Computer Vision and Pattern Recognition*, December 2001.

[56] MORTARA, M. and SPAGNUOLO, M., "Similarity measures for blending polygonal shapes," *Computers & Graphics*, vol. 25, pp. 13–27, February 2001.

[57] NAIN, D., HAKER, S., KIKINIS, R., and GRIMSON, W. E. L., "An interactive virtual endoscopy tool," in *Workshop on Interactive Medical Image Visualization and Analysis satellite symposia of MICCAI, IMIVA'01*, (Utrecht, The Netherlands), September 2001.

[58] O'BRIEN, J., BODENHEIMER, R., BROSTOW, G., and HODGINS, J., "Automatic joint parameter estimation from magnetic motion capture data," in *Graphics Interface*, pp. 53–60, May 2000.

[59] PAVLOVIC, V., REHG, J. M., CHAM, T.-J., and MURPHY, K. P., "A dynamic bayesian network approach to figure tracking using learned dynamic models," in *ICCV (1)*, pp. 94–101, 1999.

[60] PLÄNKERS, R. and FUA, P., "Articulated Soft Objects for Video-based Body Modeling," in *International Conference on Computer Vision*, (Vancouver, Canada), July 2001.

[61] PRODGER, P., *Time Stands Still: Muybridge and the Instantaneous Photography Movement*. Oxford University Press, 2003.

[62] ROSSIGNAC, J., TURK, G., GHRIST, R. W., and SZYMCZAK, A., "Multi-scale topological analysis of time-evolving shapes." NSF-CARGO Grant #0138420, 2002.

[63] SAND, P., MCMILLAN, L., and POPOVIĆ, J., "Continuous capture of skin deformation," *ACM Transactions on Graphics*, vol. 22, pp. 578–586, July 2003.

[64] SEBASTIAN, T. B., KLEIN, P. N., and KIMIA, B. B., "Recognition of shapes by editing shock graphs," in *ICCV*, pp. I: 755–762, 2001.

[65] SEITZ, S. M. and DYER, C. R., "Photorealistic scene reconstruction by voxel coloring," *International Journal of Computer Vision*, vol. 35, pp. 151–173, November/December 1999.

[66] SIDDIQI, K., BOUIX, S., TANNENBAUM, A., and ZUCKER, S. W., "Hamilton-jacobi skeletons," *International Journal of Computer Vision*, vol. 48, pp. 215–231, July/August 2002.

[67] SIDENBLADH, H., BLACK, M., and FLEET, D., "Stochastic tracking of 3d human figures using 2d image motion," in *ECCV00*, pp. II: 702–718, 2000.

[68] SLOAN, P.-P. J., III, C. F. R., and COHEN, M. F., "Shape by example," in *2001 ACM Symposium on Interactive 3D Graphics*, pp. 135–144, March 2001.

[69] TEICHMANN, M. and TELLER, S., "Assisted articulation of closed polygonal models," in *Proceeding of Eurographics Workshop on Computer Animation and Simulation 1998*, August 1998.

[70] VEDULA, S., BAKER, S., SEITZ, S., and KANADE, T., "Shape and motion carving in 6D," in *Proceedings of Computer Vision and Pattern Recognition (CVPR2000)*, pp. 592–598, June 2000.

[71] VERROUST, A. and LAZARUS, F., "Extracting skeletal curves from 3d scattered data," *The Visual Computer*, vol. 16, no. 1, 2000.

[72] VICON, *Vicon Motion Systems*. http://www.vicon.com, 2003.

[73] WADE, L. and PARENT, R. E., "Automated generation of control skeletons for use in animation," *The Visual Computer*, vol. 18, pp. 97–110, March 2002.

[74] WEIK, S. and LIEDTKE, C.-E., "Hierarchical 3d pose estimation for articulated human body models from a sequence of volume data," in *Robot Vision* (KLETTE, R., PELEG, S., and SOMMER, G., eds.), vol. 1998 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, Germany: Spinger-Verlag, 2001.

[75] ZHANG, E., MISCHAIKOW, K., and TURK, G., "Feature-based surface parameterization and texture mapping," *ACM Transactions on Graphics*, To Appear 2004.